IN THE UNITED STATES DISTRICT COURT

FOR THE DISTRICT OF DELAWARE

TRANSMETA CORPORATION, Plaintiff and Counterclaim Defendant, v. INTEL CORPORATION, Defendant and Counterclaim Plaintiff.	Civil Action No. 06-633-GMS

INTEL CORPORATION'S OPENING CLAIM CONSTRUCTION BRIEF

John W. Shaw (No. 3362)
Karen L. Pascale (No. 2903)
YOUNG CONAWAY STARGATT & TAYLOR, LLP
The Brandywine Building
1000 West Street, 17th Floor
Wilmington, DE 19801
(302) 571-6600
kpascale@ycst.com

Attorneys for Defendant Intel Corporation

OF COUNSEL:

Matthew D. Powers
Jared Bobrow
Steven S. Cherensky
WEIL, GOTSHAL & MANGES LLP
201 Redwood Shores Parkway
Redwood Shores, CA 94065
(650) 802-3000

Kevin Kudlac
WEIL, GOTSHAL & MANGES LLP
8911 Capital of Texas Highway, Suite 1350
Austin, TX 78759
(512) 349-1930

Dated: October 19, 2007

TABLE OF CONTENTS

Page(s)

TABLE OF CONTENTS
(continued)

Page(s)

TABLE OF CONTENTS
(continued)

Page(s)

TABLE OF CONTENTS
(continued)

Page(s)

TABLE OF AUTHORITIES

Page(s)

Cases

TABLE OF AUTHORITIES
(continued)

Page(s)

=

TABLE OF ABBREVIATIONS

"JA-A"	Joint Appendix Part A	
"JA-B"	Joint Appendix Part B	
"JA-C"	Joint Appendix Part C	
"JA-D"	Joint Appendix Part D	
"JA-E"	Joint Appendix Part E	
"JA-F"	Joint Appendix Part F	
'624	U.S. Patent No. 5,737,624	(JA-A at 077-97)
'526	U.S. Patent No. 5,974,526	(JA-A at 098-119)
'433	U.S. Patent No. 6,289,433	(JA-A at 120-41)
'687	U.S. Patent No. 5,493,687	(JA-A at 014-33)
'986	U.S. Patent No. 5,838,986	(JA-A at 034-55)
'449	U.S. Patent No. 6,044,449	(JA-A at 056-76)
'503	U.S. Patent No. 5,895,503	(JA-A at 142-57)
'733	U.S. Patent No. 6,226,733	(JA-A at 158-74)
'668	U.S. Patent No. 6,430,668	(JA-A at 175-88)
'699	U.S. Patent No. 6,813,699	(JA-A at 189-201)
'061	U.S. Patent No. 7,100,061	(JA-A at 001-13)
'375	U.S. Patent No. 5,745,375	(JA-D at 001-10)
'554	U.S. Patent No. 5,617,554	(JA-D at 011-34)
'605	U.S. Patent No. 5,802,605	(JA-D at 035-57)
'101	U.S. Patent No. 5,819,101	(JA-D at 058-86)
'275	U.S. Patent No. 5,881,275	(JA-D at 087-114)
'634	U.S. Patent No. 6,385,634	(JA-D at 115-35)
'529	U.S. Patent No. 6,418,529	(JA-D at 136-58)
"FH"	File History	

TABLE OF REFERENCES TO JOINT CLAIM CONSTRUCTION CHART

A. SEIKO-EPSON REGISTER RENAMING FAMILY — **Joint Claim Construction Chart Tab A, Page Number**

B. THE SEIKO EPSON MULTI-TYPE REGISTER FAMILY

TABLE OF REFERENCES TO JOINT CLAIM CONSTRUCTION CHART

Joint Claim Construction
Chart Tab A, Page Number

TABLE OF REFERENCES TO JOINT CLAIM CONSTRUCTION CHART

Joint Claim Construction
Chart Tab A, Page Number

TABLE OF REFERENCES TO JOINT CLAIM CONSTRUCTION CHART

Joint Claim Construction
Chart Tab A, Page Number

D. THE TRANSMETA POWER MANAGEMENT PATENT

"internal conditions"

"state"

"clock frequency generator"

"processor determining"

"determining . . . maximum[s and a] . . . minimum"

"determining a frequency and a voltage"

"means for detecting . . . and causing"

TABLE OF REFERENCES TO JOINT CLAIM CONSTRUCTION CHART

E. THE INTEL POWER MANAGEMENT PATENT — **Joint Claim Construction Chart Tab B, Page Number**

F. THE INTEL ADDRESS TRANSLATION FAMILY

G. THE INTEL MULTIMEDIA PROCESSING PATENTS

TABLE OF REFERENCES TO JOINT CLAIM CONSTRUCTION CHART

Joint Claim Construction
Chart Tab B, Page Number

INTRODUCTION

Defendant and counterclaim plaintiff Intel Corporation submits this Opening Claim Construction Brief pursuant to this Court's May 2, 2007 Scheduling Order.

Transmeta has asserted 172 claims from eleven patents in this litigation. These patents are from four families, as summarized in the table below. All of the patents in each family claim the same priority date and share a common specification.[1]

Asserted Transmeta Family	Patents	Asserted Claims
Seiko Epson Register Renaming Family	'624 (JA-A at 077-97)	1-4, 6-10, 12-16, 19
	'526 (JA-A at 098-119)	1-7, 13-16, 19-20, 26-29, 34
	'433 (JA-A at 120-41)	1-4, 6-10, 12-16, 19
Seiko Epson Multi-Type Register Family	'687 (JA-A at 014-33)	1
	'986 (JA-A at 034-55)	1, 11
	'449 (JA-A at 056-76)	1, 5, 7-10
Belgard Address Translation Family	'503 (JA-A at 142-57)	21-23
	'733 (JA-A at 158-74)	1-4, 17-18, 28-30, 36, 39-42, 48-74
	'668 (JA-A at 175-88)	1-5, 15-17, 20-22
	'699 (JA-A at 189-201)	1-3, 7-10, 13-15
Power Management Family	'061 (JA-A at 001-13)	1-8,[2] 15-54, 56-57

The parties dispute the meaning of 115 terms in these 172 claims. The parties' respective proposed constructions and intrinsic evidence for these terms are set out in Exhibit A to the parties' Joint Claim Construction Statement (D.I. 92). For the Court's convenience, the disputed terms and respective constructions for the four asserted Transmeta patent families are set forth in Intel Appendices A through D to this brief, which can be found in the Joint Appendix at JA-C 130-34 [Intel Appendix A], JA-C 135-40 [Intel Appendix B], JA-C 141-56 [Intel Appendix C],

[1] The '061 Transmeta Power Management patent is the only patent of the eleven that Transmeta originally filed itself. The other ten were purchased from others – six from Seiko Epson, which in turn purchased the rights to these patents from S-MOS Systems, and four from individual Richard Belgard. Transmeta does not contend that any of its Intel-compatible processors ever practiced any of the claims of the patents purchased from Seiko Epson or Mr. Belgard.

[2] Transmeta added claims 6 and 8 of the '061 patent after the parties had already exchanged claim terms to be construed.

and JA-C 157-59 [Intel Appendix D]. Intel addresses the construction of the disputed terms in the asserted Transmeta patents in Section I, *infra*.

In its counterclaims, Intel asserted seven patents against Transmeta. These Intel patents, and asserted claims, can be grouped into three families, as summarized in the table below:

Asserted Intel Family	Patents	Asserted Claims
Power Management Patent	'375 (JA-D at 001-10)	1-4, 6-7, 16-23, 30-31, 33
Address Translation Patents	'554 (JA-D at 011-34)	1-5, 19-21
	'605 (JA-D at 035-57)	1-3, 11-12
Multimedia Processing Patents	'101 (JA-D at 058-86)	1-7, 9-15
	'275 (JA-D at 087-114)	1-10
	'634 (JA-D at 115-35)	3-6, 10, 12, 14
	'529 (JA-D at 136-58)	30-31, 33-34, 36-38, 48-50

The 21 disputed terms and respective constructions for the three asserted Intel patent families are set forth in Intel Appendices E through G to this brief, which can be found in the Joint Appendix at JA-F 001 [Intel Appendix E], JA-F 002-03 [Intel Appendix F], and JA-C 004 [Intel Appendix G], and Intel addresses the construction of these terms in Section II, *infra*.

Intel is mindful of the large number of disputed claim terms that the parties are bringing to the Court, and the unusual length of this brief. But the stakes are substantial, and the differences in the approaches taken by the parties are stark. Transmeta would have this Court adopt constructions that would stretch the scope of the Transmeta Patents-in-Suit to cover subject matter well beyond what the applicants represented to have invented to the Patent Office.

As described below, the Transmeta Patents-in-Suit have gone through lengthy prosecutions where numerous disclaimers were made and many prior art references distinguished, and the specifications and prosecution histories describe particular characteristics of "the present invention." Transmeta's constructions, almost uniformly, ignore these unambiguous statements and their legal import, and seek to distort their claims like the proverbial "nose of wax" that can be twisted and turned in any direction, in an attempt to cover

Intel's accused processors, or, as they have done for 70 of the 115 disputed terms in the Transmeta Patents-in-Suit, propose that the jury be given no guidance at all, and that no construction whatsoever be adopted. Transmeta's claims should not be interpreted to cover (or be allowed to cover through no interpretation) what Transmeta (or those they purchased the patents from) did not invent, disclaimed, or distinguished in order to have the claims issued.

LEGAL FRAMEWORK

A. General Principles of Claim Construction

Claim construction is a matter of law. *Cybor Corp. v. FAS Techs., Inc.*, 138 F.3d 1448, 1454-56 (Fed. Cir. 1998) (*en banc*). "[T]he claims of a patent define the invention," but "must be read in view of the specification, of which they are a part." *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312-15 (Fed. Cir. 2005) (*en banc*).

1. The Specification Is The "Single Best Guide" To Claim Construction

The Federal Circuit rendered the *Phillips* decision *en banc* to resolve the conflict arising from two competing methodologies of claim construction: one heavily premised on the "ordinary meaning" of claim terms as evidenced by "objective resources" such as dictionary definitions; the other giving primary weight to the intrinsic record, wherein claim terms are not to be construed "in the abstract, out of [their] particular context." *Phillips*, 415 F.3d at 1319-24. The Federal Circuit's focus on determining a construction true to the invention led the court to reject the dictionary-centric methodology in favor of the contextual methodology. *Id.*

As a result, it is "entirely appropriate for a court, when conducting claim construction, to rely heavily on the written description for guidance as to the meaning of the claims," because the specification "[u]sually is dispositive; it is the single best guide to the meaning of a disputed term." *Id.* at 1315-17. The proper construction "can only be determined and confirmed with a full understanding of what the inventors actually invented and intended to envelop with the

claim. The construction that stays true to the claim language and most naturally aligns with the patent's description of the invention will be, in the end, the correct construction." *Id.* at 1316.

The claims of a patent cannot be construed more broadly than "the invention that is set forth in the specification." *On Demand Machine Corp. v. Ingram Indus., Inc.*, 442 F.3d 1331, 1340 (Fed. Cir. 2006). While it is inappropriate to limit the claims to "an example of how to practice the invention" described in the specification, it is appropriate to rely on "the manner in which the patentee uses a term within the specification" to determine whether the patentee "intends for the claims and the embodiments in the specification to be strictly coextensive." *Phillips*, 415 F.3d at 1323. In addition, the claims may be limited to an embodiment described in the specification where there is "nothing in the context to indicate that the patentee contemplated any alternative." *Id.* (citation omitted). In fact, "the specification may reveal an intentional disclaimer, or disavowal, of claim scope by the inventor." *Id.* at 1316. "Where the specification makes clear that the invention does not include a particular feature, that feature is deemed to be outside the reach of the claims of the patent, even though the language of the claims, read without reference to the specification, might be considered broad enough to encompass the feature in question." *SciMed Life Sys. v. Advanced Cardiovascular Sys.*, 242 F.3d 1337, 1341 (Fed. Cir. 2001). For instance, "repeated derogatory statements" concerning a particular aspect of the art "are the equivalent of disavowal of that subject matter from the scope of the patent's claims." *Honeywell Int'l, Inc. v. ITT Indus.*, 452 F.3d 1312, 1320 (Fed. Cir. 2006).

2. Statements Made During Prosecution May Also Limit The Claims

It is also appropriate for a court conducting claim construction to rely on the inventor's statements to the Patent Office during prosecution, which "demonstrat[e] how the inventor understood the invention and whether the inventor limited the invention in the course of prosecution, making the claim scope narrower than it would otherwise be." *Phillips*, 415 F.3d at

1317; *see also Ormco Corp. v. Align Tech., Inc.*, 498 F.3d 1307, 1316 (Fed. Cir. 2007) ("[T]o attribute to the claims a meaning broader than any indicated in the patents and their prosecution history would be to ignore the totality of the facts of the case and exalt slogans over real meaning."); *Std. Oil Co. v. Am. Cyanamid Co.*, 774 F.2d 448, 452 (Fed. Cir. 1985) ("[T]he prosecution history [ ] limits the interpretation of claims so as to exclude any interpretation that may have been disclaimed or disavowed during prosecution [ ] to obtain claim allowance.").

3. Extrinsic Evidence Is Generally "Less Reliable"

While extrinsic evidence, including dictionaries, may be useful, the Federal Circuit views "extrinsic evidence in general as less reliable." *Phillips*, 415 F.3d at 1318. As a result, extrinsic evidence should only be considered "in the context of the intrinsic evidence," *i.e.*, the claims, specification, and prosecution history. *Id.* at 1319; *see also Old Town Canoe Co. v. Confluence Holdings Corp.*, 448 F.3d 1309, 1318 (Fed. Cir. 2006) (patentee "is not entitled to a claim construction divorced from the context of the written description and prosecution history").

B. Construction of Means-Plus-Function Terms

Use of the word "means" creates a presumption that § 112, ¶ 6 applies. *See Personalized Media Comm., LLC v. Int'l Trade Comm'n*, 161 F.3d 696, 703-04 (Fed. Cir. 1998). The claim must recite "sufficient structure" to "perform entirely the recited function" to overcome this presumption. *Sage Prods. v. Devon Indus., Inc.*, 126 F.3d 1420, 1427-28 (Fed. Cir. 1997).

Construction of a means-plus-function term is a two-part process: (1) determining the claimed function and (2) identifying the structure corresponding to that function. *See JVW Enter., Inc. v. Interact Accessories, Inc.*, 424 F.3d 1324, 1330 (Fed. Cir. 2005). A structure only qualifies as corresponding structure "if the specification or prosecution history clearly links or associates that structure to the function recited in the claim." *Default Proof Credit Card Sys., Inc. v. Home Depot U.S.A., Inc.*, 412 F.3d 1291, 1297-98 (Fed. Cir. 2005).

ARGUMENT

I. THE TRANSMETA PATENTS-IN-SUIT

A. THE SEIKO EPSON REGISTER RENAMING PATENTS

1. Introduction To The Register Renaming Patents

The Register Renaming patents are based on a "reduced instruction set computer" ("RISC") processor project at a company called S-MOS Systems in the early 1990s that was never commercialized. After S-MOS filed the applications which later issued as the Seiko Epson Register Renaming patents, as well as the applications that later issued as the Multi-Type Register patents discussed below, S-MOS sold its rights in these patents and the related technology to Seiko Epson, which also never commercialized the technology. Seiko Epson sold its rights in the Register Renaming patents (and the Multi-Type Register patents) to Transmeta in 2001. Transmeta appears to have purchased the Seiko Epson patents for the sole purpose of asserting those patents against others, for Transmeta concedes that it has never practiced any of the inventions of the asserted Seiko Epson claims.

The Register Renaming patents Transmeta acquired are directed to a very specific technique for register renaming. Register renaming allows a processor to perform instructions out of program order, which can improve computer efficiency. As demonstrated below, Transmeta would have this Court adopt constructions that would distort the Register Renaming patents to cover prior art register renaming techniques that were disclaimed in the specification and distinguished during prosecution. This is improper.

In particular, the Register Renaming applicants conceded to the Patent Office that it was well known in the art to use a structure that the applicants referred to as a "temporary buffer" to perform register renaming. In order to distinguish their claimed register renaming invention using a temporary buffer from prior art temporary buffers, the applicants represented that their

claims were directed to a specific register renaming technique where the position of each instruction in a so-called "instruction window" maps one-for-one to each storage location in the temporary buffer. The applicants also explicitly contrasted their invention from what they described in the specification as "conventional register renaming." Transmeta's attempt to distort the Register Renaming patents to cover the very prior art register renaming techniques distinguished during prosecution is improper and should be rejected.

(a) The Problem Addressed By The Register Renaming Patents

The Register Renaming patents address the problem of resolving certain "dependencies" that arise when computer program instructions are performed out of order. '624 at 1:44-49. The specification explains that there are two categories of dependencies: "true" dependencies and "storage conflict" dependencies. *Id.* at 1:52-64. True dependencies occur when an instruction uses the result of a prior instruction as an input (*id.* at 8:22-24), and therefore the later instruction cannot begin until the earlier instruction has been performed. *Id.* at 1:59-62; 8:30-31. Storage conflict dependencies occur when the outputs of two different instructions are to be stored in the same place (an "output" dependency), or the output of an instruction is to be stored in the same place as the input of a prior instruction (an "anti-dependency"). *Id.* at 8:24-28. In contrast to true dependencies which cannot be eliminated, storage conflict dependencies can be eliminated by creating additional storage locations. *Id.* at 2:11-18.

The specification acknowledges that it was well known in the art to use a temporary buffer (or "reorder buffer") to create the additional storage locations required to resolve storage conflict dependencies. *Id.* at 2:20-25; 4:24-47; 6:4-15. In prior art temporary buffers, each instruction is assigned a location in the temporary buffer in program order. *Id.* at 4:24-47; JA-A at 998-1003 [Johnson, "Superscalar Microprocessor Design" ("Johnson"), pp. 48-50, 92-94]. The processor may then execute the instructions out of program order without worrying about

storage conflict dependencies. '624 at 4:24-47. After an instruction is performed, its result is written into the temporary buffer location assigned to that instruction. *Id.*

The problem addressed by the Register Renaming patents is finding where in the temporary buffer the result of a particular instruction is stored. During prosecution, the applicants criticized the prior art approaches of using a complex associative look-up or a mapping table as expensive and time consuming. *See, e.g.*, '624 at 6:51-54; JA-A at 459 ['499 FH at 9/30/94 Response, p. 10]; *id.* at 552-54 ['433 FH at 12/27/00 Reply, pp. 4-6].

(b) The Invention Claimed In The Register Renaming Patents

The specific technique that the Register Renaming patents propose to address the prior art disadvantages is mapping each instruction to a specific, predetermined location in the temporary buffer based on the instruction's position in what the patents refer to as an "instruction window":

> In accordance with the present invention, instructions are processed in buckets. Each bucket comprises a predetermined number of instructions. The temporary buffer is set equal to the predetermined size of the buckets. Thus, each instruction's outputs (execution results) are stored in a specific, predetermined location in the temporary buffer. That predetermined location is unique to one instruction while the corresponding bucket is being processed. Thus, once dependencies between instructions are determined, the location within the temporary buffer of inputs required for a next instruction are known *a priori*.

JA-A at 440 ['499 FH at 10/14/93 Response, p. 5] (emphasis added unless otherwise noted); *id.* at 460 [*id.* at 9/30/94 Response, p. 11] ("the location at which an instruction's results are stored in a temporary buffer is related to the location of that instruction in the instruction window").

As a result, the processor is able to find instruction results in the temporary buffer without a complex associative look-up or mapping table because each location in the "instruction window" maps to a specific, predetermined location in the temporary buffer. Transmeta's constructions are improper because they ignore what the applicants told the Patent Office was the heart of their invention and seek to capture the very prior art distinguished to obtain the patents.

2. The "instruction window" terms (Renaming Terms 3 and 11)

Each of the asserted claims requires an "instruction window" ('624 and '433) or an "instruction buffer" ('526). The table below shows the parties' respective constructions for "instruction window." The parties agree that "instruction buffer" refers to the physical location where the instructions in the instruction window are stored.

Term	Intel's Construction	Transmeta's Construction
"instruction window" [Renaming Term 3][3]	the group of instructions for which the computer system determines dependencies at the same time, wherein the number of instructions in the instruction window is equal in size to the number of storage locations in the temporary buffer[4]	a group of the instructions resulting from decoding that have not been retired

There are two disputes between the parties. First, the parties dispute whether the group of instructions in the window are the instructions that are compared against each other for true dependencies (Intel's position) or the instructions that have been decoded but not yet retired (Transmeta's position). Second, the parties dispute whether the instruction window is the same size as the temporary buffer (Intel's position) or can be any size (Transmeta's position).

(a) The Specification Consistently And Repeatedly Describes The Instruction Window As The Group Of Instructions Compared Against Each Other For True Dependencies

As explained above, using a temporary buffer only resolves storage conflict dependencies. When there is a true dependency between two instructions, the later instruction cannot be performed until after the earlier instruction from which it depends is performed. '624

[3] For the page in the Joint Claim Construction Chart where the construction for each term could be found, please see the Table of References to the Joint Claim Construction Chart at page viii.

[4] Intel's construction is a slight modification of the construction Intel proposed in the Joint Claim Chart to make clear that it is the number of instructions in the instruction window that is equal in size to the number of storage locations in the temporary buffer.

at 8:30-31. The claimed invention uses a "data dependency checker" ("DDC") to "locate the [true] dependencies between the instructions for a group of instructions." *Id.* at 6:36-37.

The specification consistently and repeatedly describes this "group of instructions" for which true dependencies are determined as the instructions in the "instruction window." *See, e.g.*, 3:2-4 ("the number of dependencies between a group of instructions, such as a group of instructions in a window"); 9:43-45 ("All source registers are compared with all previous destination registers for each instruction in window 102."). The instructions in the instruction window are "defined" as "current instructions" (*id.* at 8:42-44), which are also described as the group of instructions for which the computer system determines dependencies at the same time. *See, e.g., id.* at 8:31-38 (renaming circuitry is "used to locate the input dependencies between current instructions"); 9:37-45 ("DDC 108 determines where the input dependencies are between the current instructions" by comparing "[a]ll source registers" with "all previous destination registers for each instruction in window 102"). The "instruction window" therefore is the group of instructions for which the computer system determines true dependencies at the same time.

(b) The Instruction Window Must Be The Same Size As The Temporary Buffer For The One-To-One Mapping Scheme To Work

As explained above, the applicants distinguished the claimed invention from prior art temporary buffers based on how instructions are mapped to temporary buffer locations. In order for each instruction to map to a specific, predetermined location in the temporary buffer based on that instruction's position in the "instruction window," each position in the instruction window must map one-to-one with each location in the temporary buffer. During prosecution, the applicants explained that their one-to-one mapping scheme worked by setting the number of instructions in the instruction window – which includes a predetermined number of "buckets" ('624 at 7:63-8:10) – "equal" to the number of storage locations in the temporary buffer:

> In accordance with the present invention, instructions are processed in buckets. Each bucket comprises a predetermined number of instructions. The temporary buffer is set equal to the predetermined size of the buckets.

JA-A at 440 ['499 FH at 10/14/93 Response, p. 5]. *See Gaus v. Conair Corp.*, 363 F.3d 1284, 1290 (Fed. Cir. 2004) ("according to the invention" language describes "the invention itself").

As a result, the number of instructions in the instruction window must be equal in size to the number of storage locations in the temporary buffer. This interpretation is consistent with the only embodiment described in the specification, in which the instruction window includes eight instructions and the temporary buffer has eight storage locations. *See, e.g.*, '624 at 8:2-4 ("window 102 comprises eight instructions"); *id.* at 8:50-52 ("group of 8 temporary buffers").

(c) Transmeta's Construction Is Inconsistent With The Preferred Embodiment

In addition to not including the requirements set forth above, Transmeta's construction is inconsistent with the preferred embodiment. *See Chimie v. PPG Indus., Inc.*, 402 F.3d 1371, 1377 (Fed. Cir. 2005) ("As we have frequently stated, a construction that would not read on the preferred embodiment would rarely, if ever be correct.") (quotations omitted).

In the preferred embodiment, the instruction window includes two buckets of four instructions each. '624 at 7:63-8:4. The specification explains that the four instructions in each bucket remain part of the "instruction window" until all four are retired.[5] *Id.* at 8:6-10. Instructions are retired in program order; that is, the first instruction retires first, the second instruction retires second, etc. *Id.* at 8:63-9:7. This is inconsistent with Transmeta's construction, which defines the "instruction window" as the decoded instructions "that have not

[5] The specification explains that an instruction is "retired" when its result is moved from the temporary buffer to the register file. '624 at 8:56-62.

been retired," because the first three instructions in the bucket will remain part of the instruction window – even after they retire – until the fourth instruction in the bucket also retires. *Id.*

3. The "determined by" and "assigned to" terms (Renaming Terms 4, 10, 12, 15, 17, and 18)

Each of the asserted claims requires each instruction to be mapped to a location in the temporary buffer. The "determined by" and "assigned to" terms listed above specify how the instructions are mapped to temporary buffer locations. The table below shows the parties' respective constructions for these term in claim 1 of the '624 and '433 patents.

Term	Intel's Construction	Transmeta's Construction
"one of a plurality of storage locations being determined by a location of said instruction in an instruction window" [Renaming Term 4]	each instruction in the instruction window maps to a specific, predetermined location in the temporary buffer based on that instruction's position in the instruction window	storage locations are assigned based on the program order of instructions in the instruction window
"said one of said plurality of storage locations being assigned to said instruction in said instruction window" [Renaming Term 17]	each instruction in the instruction window maps to a specific, predetermined location in the temporary buffer based on that instruction's position in the instruction window	No construction – plain and ordinary meaning If the Court decides a construction is necessary, this term means one of the plurality of storage locations in the temporary buffer is assigned to the instruction in the instruction window

The dispute between the parties is whether each instruction maps to a specific, predetermined location based on its position in the instruction window (Intel's position) or is assigned a location based on its program order (Transmeta's position for the '624 patent) or in any order (Transmeta's position for the '526 and '433 patents).

As explained above, the applicants distinguished the claimed invention from prior art temporary buffers based on how instructions are mapped to temporary buffer locations. In the specification, the applicant clearly described the "present invention" as "automatically map[ping]

each instruction [to] a predetermined temporary buffer location." '624 at 6:49-52; *see also id.* at 8:52-56 ("When an instruction completes . . . its result is stored in its preassigned location.").

Likewise, during prosecution, the applicants distinguished the claimed invention from the prior art as allowing the processor to find the result of a particular instruction in the temporary buffer without a complex associative look-up or mapping table because each instruction automatically maps to a "specific, predetermined location in the temporary buffer":

> In accordance with the present invention, instructions are processed in buckets. Each bucket comprises a predetermined number of instructions. The temporary buffer is set equal to the predetermined size of the buckets. Thus, each instruction's outputs (execution results) are stored in a **specific, predetermined location** in the temporary buffer. That predetermined location is unique to one instruction while the corresponding bucket is being processed. Thus, once dependencies between instructions are determined, the location within the temporary buffer of inputs required for a next instruction are known *a priori*.

JA-A at 440 ['499 FH at 10/14/93 Response, p. 5].

The applicants explained that the specific, predetermined location in the temporary buffer for each instruction is based on that instruction's position in the instruction window:

> According to Applicants' specification, each instruction always writes to the same place in the temporary buffer. That is, the location at which an instruction's results are stored in a temporary buffer is related to the location of that instruction in the instruction window.

Id. at 460 ['499 FH at 9/30/94 Response, p. 11].

Because the applicants distinguished their invention from prior art temporary buffers – which assigned locations in the temporary buffer in program order (*see, e.g.*, JA-A at 998-1003 [Johnson, pp. 48-50, 92-94]) – based on how the instructions are mapped to temporary buffer locations, each instruction must map to a specific, predetermined location in the temporary buffer based on the instruction's position in the instruction window. *See Ormco*, 498 F.3d at 1316 ("[T]o attribute to the claims a meaning broader than any indicated in the patents and their prosecution history would be to ignore the totality of the facts of the case and exalt slogans over

real meaning."). This interpretation is consistent with the only embodiment described, in which, for example, slot 0 in the instruction window always maps to location 0 in the temporary buffer. *See, e.g.*, '624 at 13:58 ("t0=inst. 0's slot in temporary buffer").

4. The "computer system" terms (Renaming Terms 2 and 14; Multi-Type Terms 1 and 17)

Most of the asserted claims of the Register Renaming patents are directed to a "computer system" or "processor." The table below shows the parties' respective constructions for these terms. The related terms in the Multi-Type Register patents, *i.e.*, "processor" and "central processing unit," which appear in all of the Multi-Type Register asserted claims, involve the same dispute and are therefore discussed in this section as well.[6]

Term	Intel's Construction	Transmeta's Construction
"computer system" [Renaming Term 2]	a reduced instruction set computer	No construction necessary – plain and ordinary meaning
"processor" [Renaming Term 14 and Multi-Type Term 17]	a processor in a reduced instruction set computer	No construction necessary – plain and ordinary meaning

The dispute between the parties is whether the claims are limited to resolving storage conflict dependencies in a reduced instruction set computer ("RISC") (Intel's position) or also cover doing so in a complex instruction set computer ("CISC") (Transmeta's position).

For the Register Renaming patents, it is clear from the entirety of the specification that "[t]he present invention" is directed to RISC computers. The first sentence of the register renaming specification states that "the present invention" relates to "reduced instruction set computers (RISC)." '624 at 1:29-33. The first sentence of the "Summary of the Invention" section of the register renaming specification also describes "[t]he present invention [as] directed

[6] The Register Renaming and Multi-Type Register patents include overlapping named inventors and are based on the same RISC processor project discussed above.

to instruction scheduling including register renaming and instruction issuing for superscalar RISC computers." *Id.* at 6:19-21; *see Verizon Servs. Corp. v. Vonage Holdings Corp.*, -- F.3d. --, C.A. Nos. 2007-1240, -1274, 2007 U.S. App. LEXIS 22737, at *31-32 (Fed. Cir. Sept. 26, 2007) ("When a patent thus describes the features of the 'present invention' as a whole, this description limits the scope of the invention."); *Watts v. XL Sys., Inc.*, 232 F.3d 877, 882-83 (Fed. Cir. 2000) ("[T]he specification actually limits the invention" by "stating that '[t]he present invention utilizes [the varying taper angle] feature.'").

The abstract for all three Register Renaming patents also describes the invention as a "register renaming system for out-of-order execution of a set of reduced instruction set computer instructions." *See, e.g.*, '624 at Abstract. Even the titles for all three Register Renaming patents – and all three Multi-Type Register patents – include the term "RISC." *See Honeywell*, 452 F.3d at 1318 ("the public is entitled to take the patentee at his word" when he repeatedly stated that "this invention" or "present invention" was a fuel filter).

Similarly, in the Multi-Type Register patents, the applicants explicitly distinguished the claimed RISC invention from CISC processors:

> [T]he invention may be characterized as a RISC microprocessor having a register file optimally configured for use in the execution of RISC instructions, as opposed to conventional register files which are sufficient for use in the execution of CISC (complex instruction set computing) instructions by CISC processors.

'687 at 5:50-55. In so doing, the applicants disclaimed CISC processors. *See SciMed*, 242 F.3d at 1341 ("Where the specification makes clear that the invention does not include a particular feature, that feature is deemed to be outside the reach of the claims of the patent, even though the language of the claims, read without reference to the specification, might be considered broad enough to encompass the feature in question.").

As the applicants for both the Register Renaming and Multi-Type register patents explained, there is a significant distinction between a RISC and CISC computer, which is not merely a question of nomenclature but rather considerably impacts the architecture and operation of the computer. For example, in U.S. Patent No. 5,539,911 ("the '911 patent") – incorporated by reference in the Register Renaming and Multi-Type Register patents (*see* '624 patent at 1:16-24; '687 at 1:11-14) – the applicants explained why a RISC computer is very different than a CISC computer, and why the applicants limited their claimed inventions to RISC computers:

> The design of the RISC architecture generally avoids or minimizes the problems encountered by CISC architectures with regard to branches, register references and exceptions. The pipeline involved in a RISC architecture is short and optimized for speed. The shortness of the pipeline minimizes the consequences of a pipeline stall or clear as well as minimizing the problems in restoring state-of-the-machine to an earlier execution point.

JA-A at 916 ['911 patent at 3:34-41]; *see generally id.* at 915-16 [*id.* at 1:63-3:60].

5. The "register renaming" terms (Renaming Terms 1, 13, and 16)

Each of the asserted '624 and '433 claims is directed to "register renaming." The table below shows the parties' respective constructions for this term. The related "associating" terms are Renaming Terms 13 and 16 in the '526 patent.

Term	Intel's Construction	Transmeta's Construction
"register renaming" [Renaming Term 1]	removing storage conflicts without actually renaming register addresses in the instruction	naming multiple physical registers for the same architectural register in order to reduce or eliminate storage conflicts

The parties agree that "register renaming" relates to resolving storage conflict dependencies but disagree as to whether the claims include renaming register addresses in the instructions to do so. Although register renaming generally refers to creating additional storage locations to resolve storage conflict dependencies, the applicants clearly disclaimed what they

referred to as the "conventional" approach of register renaming. In the "Summary of the Invention" section, the applicants described the "present invention" as follows:

> In contrast to conventional register renaming, the present invention does not actually rename register addresses.

'624 at 6:25-27. The specification describes "conventional" register renaming as "renam[ing] the original register identifier in the instruction to identify the new register" or storage location in the temporary buffer. *Id.* at 3:24-26; *see generally id.* at 3:14-26. Rather than the so-called "conventional" approach of renaming register addresses in the instruction itself, the Register Renaming patents are directed to creating separate tags to identify the location in the temporary buffer for each operand. *See, e.g., id.* at 6:65-7:3; 7:4-16; Fig. 1.

6. "data dependency checker" (Renaming Term 6)

The parties agree that the claimed "data dependency checker" or "DDC" determines the true dependencies between the instructions in the instruction window, but disagree as to whether the logic compares each instruction in the window to each previous instruction (Intel's position) or merely compares some of the instructions (Transmeta's position). It is clear from the "Summary of the Invention" that the DDC compares the addresses of the inputs of each instruction in the window to the address of the output of each previous instruction in the window. *See, e.g.,* '624 at 6:37-40 ("The DDC [identifies dependencies] by comparing the addresses of the source registers of each instruction to the addresses of the destination registers of each previous instruction in the group."); *id.* at 9:53-55 ("All source register addresses are compared with all previous destination register addresses for the instructions in window 102.").

7. Method Claims 1, 2, And 34 Of The '526 Patent Must Be Performed In The Recited Order

The parties dispute whether the numbered steps in method claims 1, 2 and 34 of the '526 patent must be performed in the recited order. The law requires the steps of a method claim to be

performed in the order in which they are written if either the grammar of the claim language or the intrinsic record directly or implicitly requires that order. *See Altiris, Inc. v. Symantec Corp.*, 318 F.3d 1363, 1369-70 (Fed. Cir. 2003).

Both requirements are met here. It is clear from the grammar of the claims and the context of the invention that the steps of "storing a plurality of instructions in an instruction buffer" (step 1) and "assigning . . . a storage location[] to each one of said plurality of instructions in said instruction buffer" (step 2) must be performed in order. *See, e.g., E-Pass Technologies, Inc. v. 3Com Corp.*, 473 F.3d 1213, 1222 (Fed. Cir. 2007) (holding that the steps of "transferring a data set" and "storing said transferred data set" had to be performed in order).

The specification and prosecution history also compel that the instructions be stored in the instruction buffer before each instruction is assigned a location in the temporary buffer because each instruction's location in the temporary buffer is based on its position in the instruction window. Likewise, the instructions must be compared to determine which are dependent (step 3) before the input of a dependent instruction can be associated with the temporary storage location mapped to the instruction from which it depends (step 4). *See Loral Fairchild Corp. v. Sony Corp.*, 181 F.3d 1313, 1322 (Fed. Cir. 1999) ("Although not every process claim is limited to the performance of its steps in the order written, the language of the claim, the specification and the prosecution history support a limiting construction in this case.").

8. The Means-Plus-Function Terms

(a) "tag assignment means for receiving data dependency results . . . and outputting a tag . . ." (Renaming Term 9)

There are two disputes with respect to this term in claim 7 of the '624 patent. First, the parties dispute whether this term is subject to § 112, ¶ 6. The law is well-settled that the word

"means" creates a presumption that § 112, ¶ 6 applies. *See Personalized Media*, 161 F.3d at 703-04. Section 112, ¶ 6 applies because there is no structure to overcome the presumption.

Second, the parties disagree as to what constitutes the corresponding structure if the Court agrees that this term is subject to § 112, ¶ 6. The parties agree that the claimed function is "receiving data dependency results from a data dependency checker" and "outputting a tag in place of a register address . . . ." The Tag Assign Logic (TAL) 122 described at 14:55-15:33 and Figures 3 and 9 is the corresponding structure because the specification clearly links the TAL to receiving data dependency results from a data dependency checker, *see, e.g.*, '624 at 6:46 ("outputs of the [data dependency checker] go to the TAL"), and outputting a tag in place of a register address, *see, e.g.*, *id.* at 6:58-62 ("TAL will . . . output[ ] the tag").

(b) "means for passing . . . " (Renaming Term 8)

The parties agree that this term in claims 6 and 12 of the '624 patent is subject to § 112, ¶ 6, and that the claimed function is "passing said tags to read address ports of said temporary buffer for accessing said instruction execution results" but disagree as to what constitutes the corresponding structure. The Register File Port MUXes (RPMs) 124 described at 17:45-58 and in Figure 4 are the corresponding structures because the specification clearly links the RPMs to passing tags to the temporary buffer. *See, e.g.*, '624 at 17:37-44 ("The function of the RPMs is to provide a way . . . to get data out of register files. . . . RPMs 124 receive tag information . . . that are sent to a predetermined set of ports 119 of register file 117.").

(c) "means for transferring . . ." (Renaming Term 7)

The parties agree that this term in claims 2 and 8 of the '624 patent is subject to § 112, ¶ 6, and that the claimed function is "transferring the execution results in said plurality of storage locations in said temporary buffer to register file locations in order" but disagree as to what constitutes the corresponding structure. Transmeta asserts that the corresponding structure is

"the termination logic described in col. 8, ln. 52-col. 9, ln. 7." However, the patent does not describe any termination logic. Rather, the patent discloses only that transferring results is "controlled by termination logic, as should become evident to those skilled in the art." '624 at 8:59-62. The only disclosure corresponding to the "means for transferring" is therefore "termination logic, as should become evident to those skilled in the art." Intel intends to argue at the appropriate time that this does not constitute structure, and renders claims 2 and 8 indefinite.

B. THE SEIKO EPSON MULTI-TYPE REGISTER PATENTS

1. Introduction to the Multi-Type Register Patents

Transmeta acquired the Multi-Type Register patents from Seiko Epson with the Register Renaming patents, discussed above. Processors use registers to store data when executing instructions in what are called "execution units." The Multi-Type Register patents explain that prior art systems – the Intel 80486 processor is identified as exemplary – had a set of registers for storing integer data[7] and a separate set of registers to store floating point data.[8] In these prior art processors, the integer execution unit operated on integer data stored in the integer register set and the floating point execution unit operated on floating point data stored in the floating point register set. The specification describes this as wasteful of processor resources when, for example, the processor is performing exclusively integer operations and no floating point operations, because the floating point registers are unused. '687 at 2:60-3:7.

The solution described and claimed by the Multi-Type Register patents is that the floating point register set is also usable as an integer register set, and integer instructions can then

[7] The parties agree that the term "integer data" (Multi-Type Term 2) means "numeric data which is used to represent a positive or negative whole number or zero."

[8] The parties agree that the term "floating point" (Multi-Type Term 4) means "numeric data represented by a positive or negative sign, the digits in the number, and an exponent, specifying the magnitude of the number."

optionally select and operate on either integer data stored in the integer register set or integer data in the floating point register set. '687 at 3:58-64; 4:61-62; 10:34-48. This is accomplished through the use of a "field" in the instruction that optionally selects which of the register sets the integer instruction should use. Hence, the very core of the claimed invention is that an instruction is modifiable so that it can use either of the two register sets. This is in sharp contrast to the acknowledged prior art Intel processor. Transmeta's attempt to distort the Multi-Type Register patents to ignore the very heart of the claimed invention is improper and should be rejected.

2. "specifying which . . . register set" (Multi-Type Term 7)

Claim 1 of each of the patents requires that the instruction "specify[]" which register set is to be accessed. The table below shows the parties' respective constructions for this term.

Term	Intel's Construction	Transmeta's Construction
instruction includes a field specifying which of the first and second register sets is to be accessed [Multi-Type Term 7]	the instruction is modifiable so that it performs the operation utilizing either of the two register sets	No construction necessary – plain and ordinary meaning If the Court decides a construction is necessary, this term means that the field in the instruction indicates which register set is to be accessed.

The dispute between the parties is whether the instruction must be modifiable to utilize either of two register sets, as Intel proposes. Transmeta contends that no construction is necessary, and alternatively proposes that this term means that the "field in the instruction indicates which register set is to be accessed."

As explained above, the applicants distinguished their claimed invention as allowing an integer instruction to use either of two register sets. '687 at 2:55-3:7. To accomplish this, the applicants proposed using floating point registers as integer registers when "the available integer registers are inadequate to optimally . . . hold the necessary amount of integer data." *Id.* at 3:58-

62. Accordingly, the applicants teach including "register set-specifying fields in the instruction." *Id.* at 4:57-59; *see also id.* at 10:45-48 ("Each integer instruction executed . . . will specify whether each of its sources and destinations reside in register set A or register set B.").

Transmeta's proposal that this important term be provided to the jury without any construction is untenable, because it cannot be understood in the abstract. Transmeta's alternative proposal should also be rejected because it fails to include the requirement that the instruction be capable of using either of the two register sets.

3. "a field" (Multi-Type Term 6)

Claim 1 of the '687 and '986 patents requires "a field." The table below shows the parties' respective constructions for this term.

Term	Intel's Construction	Transmeta's Construction
a field [Multi-Type Term 6]	a dedicated portion of an instruction having a defined meaning	one or more bit locations in a computer instruction

The dispute between the parties is whether the term "field" as used in the multi-type register patent claims is a dedicated portion of an instruction having a defined meaning (Intel's position) or any bit of an instruction (Transmeta's proposal).

It is clear from the specification that each of the "fields" of the instruction is a dedicated portion of the instruction with a defined meaning. *Id.* at 10:16-17. The "field" required by the asserted claims is the "register-set-specifying fields in the instruction." *Id.* at 4:58-59. The purpose of this field is to specify which register set the instruction should use to store data. Intel's construction is consistent with this understanding.[9]

[9] Intel's construction is also consistent with the general usage of the term "field" in computer science. *See, e.g.*, JA-C at 006 [IEEE Standard Computer Dictionary (1990), p. 88] ("A specified area within a record, used for a particular data item."); *id.* at 00009 [Am. Heritage Dictionary (2d College Ed. 1991), p. 495 ("A region, as a set of adjacent columns on a punched card, used to consistently record related information."); *id.* at 013 [Microsoft Press Computer

Transmeta construction, in contrast, merely repeats an inherent characteristic of an instruction, *i.e.*, that it includes at least one bit. As a result, Transmeta's construction improperly removes this limitation from the claims, which already include "instructions." Further, Transmeta's proposal is inconsistent with the preferred embodiment. *See Chimie*, 402 F.3d at 1377. As shown in Fig. 7, the term "field" is not used to refer to any arbitrary bit within an instruction as Transmeta contends, but instead refers to a dedicated portion of the instruction having a defined meaning. For example, the register-set-specifying field has a different meaning and is located in a different portion of the instruction from the opcode field. Transmeta seeks to blur the lines between these elements to merely require a single bit somewhere in an instruction. However, the "field" is a crucial aspect of the invention, as explained in the specification, and as further demonstrated by its inclusion as an express claim limitation. Transmeta's improper attempt to delete this element of the invention from the claims should be rejected.

4. "first registers" (Multi-Type Term 3)

Claim 1 of each of the patents requires "first registers each for holding [the] integer data." The table below shows the parties' respective constructions for this term.

Term	Intel's Construction	Transmeta's Construction
first registers each for holding [the] integer data [Multi-Type Term 3]	storage locations identifiable by instructions and capable of storing at least integer data	storage locations identifiable by instructions and capable of storing only integer data

The dispute between the parties is whether the "first registers" should be construed inclusively to require at least integer data (Intel's position) or as a negative limitation to require integer data and not any other data type (Transmeta's position).

Dictionary (1991), p. 144] ("A location in a record in which a particular type of data is stored . . . . Individual fields have their own specifications as to maximum length and the type of data . . . that can be placed in them."); *id.* at 016 [Cambridge Dictionary of Sci. and Tech. (1992), p. 339] ("Predetermined section of a record.").

Proper construction of this term starts with the language of the claims. Each of the asserted claims uses "[t]he transitional term 'comprising' . . . [which] is inclusive or open-ended and does not exclude additional, unrecited elements." *See Georgia-Pacific Corp. v. U.S. Gypsum Co.*, 195 F.3d 1322, 1327-28 (Fed. Cir. 1999). Transmeta's proposal to exclude data other than integer data violates this long-standing rule of construction. Such a negative limitation must find "anchor in the explicit claim language." *Omega Eng'g v. Raytek Corp.*, 334 F.3d 1314, 1322 (Fed. Cir. 2003).

Nor does the specification support Transmeta's proposal to exclude any type of data other than integer data. This is a negative limitation because it is an attempt to exclude the universe of other types of data by claiming what the invention is not instead of what the invention is. The Manual of Patent Examining Procedure ("MPEP") explains that a "negative limitation or exclusionary proviso must have basis in the original disclosure." JA-C at 018 [MPEP § 2173.05(i)]; *see also id.* ("The mere absence of a positive recitation is not basis for an exclusion."). There is no basis to limit the first register set to storing integer data only. The specification discloses storing integer data in two different register sets, but never states that the first set be limited to storing integer data only.

5. "second registers" (Multi-Type Term 5)

Claim 1 of the '687 and '986 patents also requires "second registers each for holding the integer data and for holding floating point data." The table below shows the parties' respective constructions for this term.

Term	Intel's Construction	Transmeta's Construction
second registers each for holding the integer data and for holding floating point data [Multi-Type Term 5]	storage locations identifiable by instructions and capable of storing at least integer data and floating point data	each second register can hold integer data and, alternatively, can hold floating point data

The dispute between the parties is whether the ”second registers” should be construed consistently with how both parties propose construing the “first registers,” *i.e.*, as “storage locations identifiable by instructions and capable of storing” certain types of data (Intel’s position) or entirely differently (Transmeta’s position). In particular, Transmeta’s proposal leaves out the requirement that the storage locations are identifiable by instructions. In addition, Transmeta uses open-ended language for the second register set, but inconsistently specifies closed-ended language for the first register set. It is unclear what Transmeta seeks to include or exclude, and the difference would be confusing to the jury, and therefore, improper.

6. The “processor” terms (Multi-Type Terms 1 and 17)

These terms are discussed above with the related terms in the Register Renaming patents. *See* section I.A.4, *supra*. As explained above, it is clear that the applicants directed the inventions claimed in the Multi-Type Register and Register Renaming patents, which include overlapping named inventors and were all acquired from Seiko Epson, to RISC computers.

7. The “Boolean” terms

Claim 11 of the ’986 patent is directed to performing “Boolean” instructions.

(a) “Boolean combinational instructions” (Multi-Type Term 15)

The dispute between the parties is whether the Boolean claim is limited to the Boolean combinational instructions defined in the specification (Intel’s proposal) or covers any Boolean instruction (Transmeta’s proposal). The specification defines two different types of Boolean instructions: “Boolean combinational instructions” and “Boolean comparison instructions”:

> Boolean comparison instructions specify particular integer or floating point registers for source data to be compared, and specify a particular Boolean register for the result, so there are no dedicated, fixed-location status flags. Boolean combinational instructions combine specified Boolean registers, for performing complex Boolean comparisons without intervening conditional branch instructions, to minimize pipeline disruption.

'687 at Abstract. "Boolean combinational instructions" are therefore defined as instructions that combine the results of previous "Boolean comparison instructions." Only Intel's construction for "Boolean combination instructions" is consistent with this definition.

(b) "Boolean execution unit" (Multi-Type Term 14)

The "Boolean execution unit" corresponds to the "Boolean functional unit 70" described in the specification which, in contrast to the integer and floating point units 66 and 68 which perform comparison instructions, is "only used in performing bitwise logical combination of Boolean register contents." '687 at 12:3-8. Transmeta's proposal that this term covers any circuitry that generates Boolean results is therefore inconsistent with the patents.

(c) "Boolean result" (Multi-Type Term 16)

The dispute between the parties is whether a "Boolean result" is the single-bit result of a Boolean operation (Intel's position) or may include more than one bit (Transmeta's proposal). Only Intel's proposal is consistent with the specification, which never discloses a multi-bit result for a Boolean operation.[10] In the preferred and only embodiment described in the specification, "each Boolean register is one bit wide, indicating one Boolean value" ('687 at 8:31-33), and every Boolean combinational instruction shown in Table 2 produces a single-bit output. *Id.* at 12:15-27. The patents do not support a broader interpretation. *See Phillips*, 415 F.3d at 1323 (claims may be limited to an embodiment in the specification where there is "nothing in the context to indicate that the patentee contemplated any alternative") (citation omitted).[11]

[10] Intel's construction is also consistent with the general usage of the term Boolean. *See, e.g.*, JA-C at 005 [IEEE Standard Computer Dictionary (1990), p. 31] ("Any operation in which . . . the result takes one of two values."); *id.* at 012 [Microsoft Press Computer Dictionary (1991), p. 42] ("Two important aspects of Boolean algebra are that variables can be restricted to one of only two values, true or false.").

[11] Intel's construction also is supported by the prior sworn testimony of Richard Belgard (the named inventor on four of Transmeta's patents and a paid consultant to Transmeta). JA-C at 021

8. The "execution unit" terms

Claim 1 of the '449 patent is directed to an "execution unit."

(a) "execution unit . . . accesses" (Multi-Type Term 19)

The dispute between the parties is whether this term should be construed. It is clear that Transmeta's intent in proposing that this term not be construed (and not offering an alternative construction) is to render this term meaningless. This would be inconsistent with the claim language which explicitly recites three different activities: accessing, reading, and writing.

(b) "execution unit . . . reads" and "execution unit . . . writes" (Multi-Type Terms 20 and 21)

There are two disputes with respect to these terms. First, the parties dispute whether these terms require the instruction to be able to optionally select from either of two register sets interchangeably, as Intel proposes. For the reasons set forth above, Transmeta's attempt to read out this requirement should be rejected. Second, the parties dispute whether the "execution unit . . . writes" term requires that the execution unit have the ability to write a result to the opposite register set from which it read source data, as Transmeta proposes. There is no support for this additional requirement in the claim language or specification.

9. The Means-Plus-Function Terms

For each of the terms discussed in this section, the parties dispute whether the term is subject to § 112, ¶ 6, as Intel proposes. Each of these terms is subject to § 112, ¶ 6 because the asserted claims do not recite structure sufficient to overcome the presumption. *See Personalized Media*, 161 F.3d at 703-04 ("means" creates a presumption that § 112, ¶ 6 applies); *Sage Prods.*, 126 F.3d at 1427-28 (to rebut the presumption, the claim must "elaborate sufficient structure . . .

[Belgard 3/10/05 Tr. at 110:3-4] ("a Boolean operation is an operation that returns one bit of precision in its answer"); *id.* [*id.* at 113:2-4] ("And that [Boolean] operation gives you a true or false. There's only two answers: true or false. That's the universe of answers.").

within the claim itself to perform entirely the recited function"). With respect to "means . . . for accessing," Transmeta's reliance on the very same structure it contends corresponds to the "reading means" and "writing means" recited in the claim does not qualify as sufficient to overcome the presumption that § 112, ¶ 6 applies because it does not "perform entirely" the recited accessing function, which is necessarily broader than the combination of "reading means" and "writing means." In addition, if the Court agrees that the "reading means" and "writing means" are subject to § 112, ¶ 6, then "means . . . for accessing" is necessarily also subject to § 112, ¶ 6 because both the "reading means" and "writing means" are by definition functional.

(a) The "reading means" and "writing means" terms (Multi-Type Terms 9, 10, 12 and 13)

If the Court agrees that these terms, which appear in claim 1 of the '687 and '986 patents ("reading means" and "writing means") and claim 11 of the '986 patent ("read access means" and "write access means"), are subject to § 112, ¶ 6, then there remain disputes between the parties regarding what constitutes the recited function and the corresponding structure.

First, with respect to the recited function, the functions proposed by Transmeta for the '687 and '986 patents are inconsistent with the specification because they fail to include the requirement that integer instructions are able to select from either of two register sets.

Second, with respect to corresponding structure, the parties agree that switching and multiplexing control (SMC) units are necessary to perform the recited functions but disagree as to whether these terms are limited to the specific elements of the SMC units described in the specification for performing these functions (Intel's position) or any structures (Transmeta's position). The patents state that "[a]ll elements of FIG. 2 except the register set 20 and the elements 102-108 comprise the SMC unit B of FIG. 1" and "[a]ll elements of FIG. 3 except register set 18 and the elements 140-146 comprise the SMC unit A of FIG. 1." '687 at 15:7-8,

16:22-24. It is those structures that relate to the recited functions (*i.e.*, the 12 multiplexers labeled S1, S2 in Figs. 2A and 3A, which are controlled by instruction bits B1 and B2).[12]

(b) "means . . . for accessing" (Multi-Type Term 8)

If the Court agrees that this term, which appears in claim 1 of the '687 and '986 patents, is subject to § 112, ¶ 6, then the parties agree that the claimed function is "accessing the first register set or the second register set as specified by the field," but disagree as to what constitutes the corresponding structure. Transmeta proposes that the corresponding structure is the same structure corresponding to the "reading means" and "writing means" claim terms discussed below. Transmeta's construction is improper because construing this term as completely coextensive with the separately recited "reading means" and "writing means" would render this term meaningless. *Bicon, Inc. v. Straumann Co.*, 441 F.3d 945, 949-50 (Fed. Cir. 2006) ("claims are interpreted with an eye toward giving effect to all terms in the claim").

Because the claims recites that the "means . . . for accessing" include the "reading means" and "writing means," and because the term "include" is open-ended and not limiting, the scope of the "means . . . for accessing" term is necessarily broader than the combination of the "reading means" and "writing means. Moreover, if the "means . . . for accessing" consisted solely of the "reading means" and the "writing means," there would have been no reason for the applicants to require, in addition, that the claimed invention include a "means . . . for accessing." Transmeta's construction, which equates (and makes coextensive) the "means . . . for accessing" with the "reading means" and the "writing means" makes the phrase "means . . . for accessing" meaningless." This is consistent with claim 1 of the '449 patent, which is directed to three distinct functions: reading, writing and accessing.

[12] Intel does not concede that these structures are sufficient to perform the recited functions and reserves the right to present defenses based on § 112 at the appropriate stage of this litigation.

As a result, the corresponding structure must include some structure in addition to whatever structure the Court finds corresponds to the "reading means" and "writing means" terms discussed above. Transmeta has not identified any such structure in the specification. If the Court does not include any structure in addition to whatever structure the Court finds corresponds to the "reading means" and "writing means," then Intel intends to argue at the appropriate time that claim 1 of the '687 patent and claim 1 of the '986 patent are indefinite.

C. THE BELGARD ADDRESS TRANSLATION PATENTS

1. Introduction To The Belgard Address Translation Patents

The four asserted Belgard patents, which Transmeta acquired in 2001 but never used in any of its products, were procured over a nine-year period. The patents relate to the field of address translation, which is the translation of a "virtual" memory address, used internally by a software program to identify information, into a "physical address" that specifies the actual location in the computer's physical memory where the information is stored. The patents explain that prior art techniques – such as those invented by Intel – translated a "virtual" address to a "physical" address in a two-step process. '503 at 2:60-67. The patents claim "an improved address translation method" that uses a faster one-step process.

Two features are critical to the claimed invention, both of which Transmeta is seeking to exclude from the asserted claims. First, the invention uses a modified "segment descriptor" memory to store prior "physical address" information. Although this core feature is reflected in claims of the parent patent, over the years numerous continuation patents issued, each of which moved further away from the descriptions of the invention in the specification and prosecution history. Transmeta's departure from the core of Mr. Belgard's alleged invention is demonstrated, for example, by its exclusion from this litigation of the claims that refer explicitly to the special "segment descriptor" memory that Mr. Belgard repeatedly described as the key

aspect of his claimed invention. *See, e.g.*, '503 claims 1-20; '733 claims 23-27; '668 claims 7-14. However, Transmeta cannot hide from Mr. Belgard's claimed invention, because, under governing law, each and every of the asserted claims requires the special "segment descriptor" memory at the heart of Mr. Belgard's claimed invention. *See Ormco*, 498 F.3d at 1316.

The second critical feature of Mr. Belgard's invention is the so-called "fast physical address," a coined claim term having no customary meaning in the art. This address is a guess at the memory location that needs to be accessed based on the location of the prior access, and, as a result of the special "segment descriptor" memory described above, Mr. Belgard claims that it can be generated more quickly than is possible using a prior art method. The key characteristic of the "fast physical address" is that it has the same type and amount of information as a conventional physical address so that it can access memory in exactly the same way. Indeed, the specification makes clear that Mr. Belgard's technique was compatible with existing memory interface units, and, in the prosecution history, Mr. Belgard distinguished his invention over prior art techniques that only used partial address information to access memory. Nevertheless, Transmeta now proposes overly broad constructions for "fast physical address" – and other vague, coined terms – that ignore the core of Mr. Belgard's claimed invention in an attempt to improperly extend the scope of the patents to cover the very features Mr. Belgard disclaimed.

(a) The Prior Art: Two-Step Address Translation

According to the patents, conventional address translation methods are too slow because a two-step process is required. These two steps are called (1) segmentation and (2) paging.

During the segmentation step, the virtual address is converted into an intermediate address, referred to as a "linear address." The linear address space is divided into "segments," each of which consists of a contiguous group of linear addresses and corresponds to a specific portion of a software program's data. The virtual address contains two parts: (1) a segment

identifier and (2) a segment offset. The segment identifier points to a location in a "segment descriptor" memory that keeps track of the first linear address for each segment (referred to as a "base address"). The segment offset identifies a particular linear address within the segment. To generate the linear address, the segment offset is added to the base address.

During the paging step, the intermediate linear address generated during the segmentation step is converted into the physical address. The linear address contains two parts: (1) a page number and (2) a page offset. The page number is used to access a small memory called a "page cache" that stores the physical location of a fixed-size chunk of physical memory, referred to as a "page frame." The page offset, which requires no further translation, identifies a particular location within that page frame. The location of the page frame is combined with the page offset to form the physical address used to access the physical memory.

(b) The Claimed Invention: One-Step Address Translation

The Belgard patents purport to improve the speed of this conventional address translation process by providing a "fast" address translation that eliminates the paging step discussed above. Key to this technique is a special augmented "segment descriptor" memory that stores not only the first linear address in each segment, as described above, but also a "last page frame field." This field, as its name suggests, holds the location of the page frame from the immediately prior-in-time address translation. '503 at 3:58-61. This "fast page frame" is then used to generate a "fast physical address" that, like a regular "physical address," unambiguously specifies the location of a unit of data in the main memory.

Since the "fast page frame" is derived directly from the "segment descriptor" memory using the segment identifier from the virtual address, there is no need to perform the paging step. Eliminating the paging step – made possible by the augmented "segment descriptor" memory – is the essence of the alleged invention. However, because the "fast physical address" is based on

the potentially incorrect assumption that the current memory request is for data located in the last page frame accessed, it is only a speculative address, and errors arising from its use may need to be corrected later on. This is the price paid for not having to perform the paging step.

2. Each Of The Asserted Claims Should Be Construed To Require A Page Frame Memory Accessible Based On Virtual Address Information Alone

The use of a special memory (*i.e.*, the segment descriptor memory described above) from which prior "page frames" can be retrieved solely on the basis of virtual address information is central to the Belgard patents. The specification repeatedly makes this point, explaining that this is the feature that drives the invention and distinguishes it from the prior art:

> [D]elays associated with . . . translation could be significant in overall system performance. With the recognition of this consideration, the present invention includes page information in the segment translation process. The present invention recognizes the performance penalty of the prior art and alleviates it by storing paging information in the segmentation unit obtained from a paging unit in previous linear-to-real address translations.

'503 at 6:44-52; *see also, e.g.*, *id.* at Fig. 2; 3:51-57; 6:8-14; 7:5-16.

The prosecution histories underscore the importance of retrieving prior page frames based on virtual address information alone. Indeed, during prosecution, Mr. Belgard personally explained this to the examiner:

> Mr. Belgard gave an overview of the invention, points out key elements which distinguishes the invention from that of Crawford and Hinton. The key element mentioned is that the prior arts do not include physical page address information in the segment descriptor.

JA-A at 569-70 ['503 FH at 3/6/97 Interview Summary]. Mr. Belgard continued to make similar arguments during prosecution.[13]

[13] *See, e.g.*, JA-A at 556-58 ['503 FH at 10/7/96 Response, pp. 1-3]; *id.* at 596 ['466 FH at 8/4/97 Preliminary Amendment, p. 6]; *id.* at 603-06, 608-09 ['466 FH at 11/24/98 Response, pp. 5-8, 10-11]; *id.* at 611 ['466 FH at 3/18/99 Letter, p. 1]; *id.* at 621-22 ['466 FH at 3/24/99 Response, pp. 5-6]; *id.* at 661, 663 ['733 FH at 11/24/98 Response, p. 11, 13]; *id.* at 720-21, 727 [3/20/00 Appeals Brief, p. 12-13, 19]; *id.* at 770-71 ['699 FH at 12/15/03 Response, pp. 9-10].

It is thus clear that one of skill in the art would understand the claims of the address translation patents to require the use of a memory for storing page frames that can be accessed based on virtual address information alone. Under these circumstances, the Federal Circuit has held that patent claims should be construed to require a certain limitation, even if it is not explicitly set forth in the claims. In fact, only a few months ago in *Ormco*, the Federal Circuit so held, because, like here, the "situation . . . involve[d] specifications that in all respects tell us what the claims mean, buttressed by statements made during prosecution in order to overcome a rejection over prior art." 498 F.3d at 1316. As the Federal Circuit explained, and as is the case here, "to attribute to the claims a meaning broader than any indicated in the patents and their prosecution history would be to ignore the totality of the facts of the case and exalt slogans over real meaning." *Id*; *see also Alloc, Inc. v. ITC*, 342 F.3d 1361, 1370 (Fed. Cir. 2003) ("[W]here the specification makes clear at various points that the claimed invention is narrower than the claim language might imply, it is entirely permissible and proper to limit the claims."). Accordingly, each of the asserted claims of the Belgard patents should be construed to require a page frame memory accessible by virtual address information alone.[14]

3. The "memory access" terms (Belgard Terms 11, 21, 32, 37, 42, 46, 61, 77, and 79)

Claim 21 of the '503 patent, claims 2, 17, 18, 28, 36, 40, 48, 50, 64, and 70 of the '733 patent, claims 16 and 20 of the '668 patent, and claims 2 and 10 of the '699 patent include one of the "memory access" terms listed above. The table below shows the parties' respective constructions for "fast memory reference" ('733 claim 28).

[14] The claim language also provides sufficient basis upon which this construction may be grounded. For example, the "memory" and "storing" terms refer directly to this special memory or the act of storing page frames in such a memory. Likewise, the "fast page frame" terms are inextricably connected to such a memory because they must be stored there to form a "fast physical address." Because the "fast physical address" is constructed directly from the "fast page frame," it too must be understood to require this special memory.

Term	Intel's Construction	Transmeta's Construction
"fast memory reference" [Belgard Term 37]	using a "fast physical address" to refer to memory in the same manner as if the fully translated "physical address" was already available[15]	No construction necessary – plain and ordinary meaning If the Court decides a construction is necessary, this term means using the fast physical address to locate data in memory.

The dispute between the parties is whether the "fast physical address" must be able to locate data in main memory in the same way in which the regular "physical address" locates data in main memory (Intel's position) or whether the "fast physical address" can locate data in a different way (Transmeta's position).

The Belgard patents are not directed to the process of using a physical address to access the physical memory (referred to as a "memory access"). In fact, the specification explicitly states that this is "ancillary" to the present invention and otherwise "described in the prior art." *See* '503 at 8:29-39. The specification further explains that "the memory subsystem employing the present invention" is "desirably isolated from . . . the address translation mechanism" through the use of a "conventional" "prior art" interface unit. *Id.* Mr. Belgard even distinguished prior art that was allegedly directed to memory access operations on the basis of that being a subject matter unrelated to that of the patent. *See, e.g.*, JA-A at 674 ['733 FH at 6/16/99 Letter, p. 1].

Because the claimed invention is directed to a method for generating a complete physical address more quickly than prior art methods, the "fast physical address" must be able to access memory in the same standard manner as a regular physical address generated using the prior art two-step process. This is confirmed by the specification, in which all the figures and text describe both the "physical address" and "fast physical address" as accessing memory in the

[15] Intel's construction for these terms is a more concise modification of the construction Intel proposed in the Joint Claim Chart.

same way by being sent to the exact same memory interface unit. *See, e.g.*, '503 at 9:21-24; 10:57-67; Fig. 3A-C; *see also* JA-A at 720-21 ['733 FH at 3/20/2000 Appeals Brief, pp. 12-13].

4. The "physical address" and "fast physical address" terms

The "physical address" and "fast physical address" terms listed above refer to memory addresses generated by the prior art two-step method and the "fast" method claimed in the Belgard patents, respectively. The table below shows the parties' respective constructions for "physical address" ('503 claim 21, '733 claims 49, 57, and 69, and '699 claims 7, 10, and 13) and "fast physical address" ('733 claims 1, 17, 18, 28, 36, 39, and 63, and '668 claim 15).

Term	Intel's Construction	Transmeta's Construction
"physical address" [Belgard Term 1]	an address that is sufficient to unambiguously specify the location of a desired unit of data equal in size to the smallest storage location addressable by the processor, typically one byte	a location in the computer's physical, *i.e.*, real, memory
"fast physical address" [Belgard Term 20]	an address sufficient to unambiguously specify the location of a unit of data equal in size to the smallest storage location addressable by the processor that may or may not be the desired unit of data, and which is generated quicker than a "physical address"	an address specifying the location of data that may or may not be the desired location, and which is available sooner than an actual physical address

The parties' dispute centers on whether these terms refer to an actual memory address that specifies the location of a fundamental unit of memory (Intel's position) or something else (Transmeta's position). In the overwhelming majority of processors, including the accused products, the fundamental unit of memory is the byte (*i.e.*, 8 bits). To be clear, the dispute has nothing to do with the precise size of the fundamental memory unit in the accused products (*i.e.*, the smallest storage location addressable by the computer). Rather the dispute is about whether the terms "physical address" or "fast physical address" may, as Transmeta contends, refer to an

entity that is incapable of unambiguously defining the location of a fundamental memory unit, such as a partial memory address that can be used to locate only a block of thousands of fundamental memory units or, even worse, thousands of scattered memory units. Every description of these terms, whether in the specification, claims, or prosecution history, makes clear that a "physical address" or "fast physical address" is a complete address precise enough to specify the location of a fundamental unit of memory (also referred to as a "byte" for simplicity).

(a) The "physical address" terms (Belgard Terms 1, 16, 25, 44, and 59)

As an initial matter, the specification clearly treats the byte as the fundamental unit of memory. *See, e.g.*, '503 at 1:57-58 (noting that the virtual address identifies a "byte in [a] segment"); *id.* at 2:47-49 (segments can start on "any byte boundary" and have "any byte length"); *id.* at 5:51-55 (the linear address is frequently a "byte offset"). Naturally then, the specification describes the "physical address" as an entity specifying the location of a byte. For example, Figures 1 and 3A-C of the Belgard patents all characterize the "physical address" as composed of a "page frame" and a "page offset." Verifying that this address carries enough precision to locate a single byte, the specification explains that the "page displacement field," which is the same as the "page offset," "locates a byte within the selected page frame." *Id.* at 6:2-3; *see also id.* at 2:4-6 (the "page offset" is "typically a byte offset").[16]

[16] Similarly, U.S. Patent No. 5,321,836, which is incorporated by reference, explains that a "20-bit base word . . . is combined with the 12-bit displacement field . . . and the resultant physical address selects from a 4k byte page frame in main memory." JA-A at 862 ['836 patent at 5:39-43; 5:58-63]. Given a "page frame" size of "4k byte[s]," the "12-bit displacement field" carries enough precision to specify the location of the smallest memory location addressable by the processor (*i.e.*, a byte). The above demonstrates that a "physical address" must carry enough information to specify the location of a byte of data in main memory.

(b) The "fast physical address" terms (Belgard Terms 10, 20, 36, 45, 64, 80, and 86)

As an initial matter, the construction of coined terms such as "fast physical address" and "speculative physical address" requires the guidance of the specification. *See J.T. Eaton & Co. v. Atlantic Paste & Glue Co.*, 106 F.3d 1563, 1568 (Fed. Cir. 1997) (because "[the disputed claim term] is a term with no previous meaning to those of ordinary skill in the art[,] its meaning, then, must be found somewhere in the patent."). It is clear from the specification that there is a structural identity between a full "physical address" and "fast physical address." Indeed, the Belgard patents describe a procedure for rapidly generating a "fast physical address" that is intended to be identical to the "physical address." This "fast physical address" must be able to access memory in exactly the same way as the "physical address." No other techniques are disclosed in the Belgard patents. *See* section I.C.3, *supra*. This alone demonstrates that the "fast physical address" is structurally identical to a regular "physical address," which, as explained above, must unambiguously locate a byte of data. In fact, when summarizing the formation of a "fast physical address," the specification explicitly describes it as a "full physical address" consisting of a "page frame number and page offset." '503 at 3:58-67. As explained above, the "page offset" identifies a byte within a page, thus demonstrating that the resulting "fast physical address," like a normal "physical address," unambiguously specifies the location of a byte. *See also* '503 at 9:16-21 (describing the composition of the "fast physical address" as including a page frame and page offset); *id.* at 11:34-40 (same).

The prosecution history is in accord with the evidence set forth above. Indeed, in distinguishing the prior art Toy reference, which disclosed a system for generating a partial "real address" consisting of a full offset portion and only a partial page portion, Mr. Belgard explained that such an entity was "neither a virtual, linear or physical address, nor a translated version of

any of these" and that it was "difficult to even see how Toy is even 'generating' a fast physical address." JA-A at 723 ['733 FH at 3/20/00 Appeals Brief, p. 15]. This disclaimer backs up what is already clear from the specification: a "fast physical address" is a complete address that unambiguously identifies the location of a single byte of data.

5. The "fast page frame" terms (Belgard Terms 4, 12, 23, 29, 34, 39, 41, 43, 47, 49, 50, 51, 58, 60, 63, 67, 68, 73, 74, 78, 81, 85, and 88)

The "fast page frame" terms listed above refer to portions of "fast physical addresses" that specify the location of a page of data. The table below shows the parties' respective constructions for the term "information from the first address translation" ('733 claims 51 and 57), which should be understood to be the same thing as a "fast page frame" because it too refers to the information stored in the segment descriptor memory from the prior address translation that is used to generate the "fast physical address."

Term	Intel's Construction	Transmeta's Construction
"information from the first address translation" [Belgard Term 47]	a "page frame field" (as construed herein) that may or may not specify the location of the desired page of data and that is obtained from the "physical address" (as construed herein) used in the previous request for data from the segment from which data is currently being requested	No construction necessary – plain and ordinary meaning. If the Court decides a construction is necessary, this term means at least a portion of an address translated during the first address translation

The dispute between the parties is first, whether construction of terms in this category is necessary to assist the jury (Intel's position) or not (Transmeta's position), and second, whether the ambiguity in Transmeta's alternative constructions is appropriate (Intel contends it is not).

As in the case of the "fast physical address" terms, terms such as "fast page frame" and "information from the first address translation" have no clear meaning to one of skill in the art, and the specification thus provides the best guidance as to their meaning. The specification makes clear that these terms refer to entities that are intended to be identical to the actual "page

frame" normally required for a given memory request. Indeed, the specification repeatedly described these terms as being combined with a "page offset" to form a "fast physical address," *see, e.g.*, '503 at 6:53-60; 7:5-16; 8:59-65; 10:57-67; 11:3-16; 12:14-21, which is then shipped to the exact same memory bus as the actual "physical address." *See* section I.C.3, *supra*. This alone demonstrates that a "fast page frame," like an actual "page frame," must at least contain enough information to unambiguously specify the location of a page. Otherwise it could not step into the shoes of an actual "page frame" in a "physical address."

Intel's construction further requires that the "fast page frame" be the last page frame (*i.e.*, be obtained from the previous request for data in the segment from which data is currently being requested). This is the very purpose for the augmented "segment descriptor" memory, described above, which Mr. Belgard repeatedly and forcefully explained was so central to his alleged invention. *See* section I.C.1, *supra*. This is because the "page frames" stored in the "segment descriptor memory" are undeniably necessary to form the "fast physical address." *See, e.g.*, '503 at 3:39-61; 7:5-16; 8:59-65; 10:57-67; 11:3-16; 11:34-40; 12:14-21; Fig. 3A-C. And, the specification confirms that information stored in the "segment descriptor" memory derives from prior translations "on a segment-by-segment basis." *Id.* at 3:58-61; *see also id.* at 3:39-48; 5:34-38; 7:66-8:3; JA at 00557 ['503 FH at 10/7/96 Response, p. 2]. In fact, the title of the '503 parent patent even states that the invention is an "Address Translation Method . . . Using Physical Address Information Including During A Segmentation Process." Transmeta's alternative construction for the "fast page frame" terms turns a blind eye to these requirements.

6. The "fast page offset" terms (Belgard Terms 24, 30, 33A, 33B, 38, 40, 48, 57, 69, 75, and 83)

Most of the asserted claims include one of the "fast page offset" terms listed above. The table below shows the parties' respective constructions for "partial linear address" ('699 claim

1), which should be understood to be the same as a "fast page offset" because it too refers to the page offset that is used to generate a "fast physical address."

Term	Intel's Construction	Transmeta's Construction
"partial linear address" [Belgard Term 38]	a portion of a "physical address" sufficient to unambiguously specify the location of a byte of data within a page and that is used together with a "fast page frame" to form a "fast physical address"	No construction necessary – plan and ordinary meaning. If the Court decides a construction is necessary, this term means a portion of a linear address.

The dispute between the parties is first, whether construction of terms in this category is necessary to assist the jury (Intel's position) or not (Transmeta's position), and second, whether the ambiguity in Transmeta's alternative constructions is appropriate (Intel contends it is not).

The patents say nothing about how any portion of a linear address could be used, on its own, to generate a memory address without a page frame field. Intel's construction thus makes clear that a "partial linear address" can only refer to a true "page offset" that is used in conjunction with a page frame to generate a "fast physical address." *See, e.g.*, '503 at Fig 3A-C; *id.* at 9:2-6; *id.* at 9:9-13; *id.* at 11:34-40.

D. THE TRANSMETA POWER MANAGEMENT PATENT

1. Introduction To The Power Management Patent

The '061 patent – the only patent asserted by Transmeta based on work done at Transmeta and the only Transmeta Patent-in-Suit that Transmeta claims to have practiced in its processors – is directed to managing power consumption in a computer processor by increasing or reducing voltage and/or frequency. As explained in the "Background of the Invention" section, increasing voltage and/or frequency increases power consumption, and decreasing voltage and/or frequency decreases power consumption. '061 at 1:39-47. The process of changing voltage and frequency to manage power consumption is known as "dynamic voltage-frequency scaling." *Id.* at 1:56-57.

Transmeta acknowledged that it was a latecomer to this crowded field when it applied for this patent in 2000. *Id.* at 1:56-58 ("Various methods of implementing this dynamic voltage-frequency scaling have been described in the prior art."). The '061 patent issued six years later, after a lengthy and tortured prosecution history that included six Office Actions rejecting all claims, two interviews, and numerous narrowing amendments and arguments. Time and time again, Transmeta was forced to re-define and narrow its claimed invention in response to the examiner's rejection of claims based on the prior art, including Intel U.S. Patent No. 5,812,860 to Horden *et al.* ("Horden"). In particular, Transmeta expressly distinguished power management techniques that monitor core utilization, use generic clock frequency generators, or use the operating system to control power management.

This is the final, and perhaps most compelling, example of Transmeta's strategy to recover subject matter in this lawsuit that Transmeta and the other applicants were forced to give up during prosecution to distinguish the prior art. Transmeta's improper attempts to gain property rights to which it is not entitled should be rejected.

2. "internal conditions" (Power Management Terms 5 and 10)

Claims 15 and 39 include limitations relating to "internal" processor conditions. The table below shows the parties' respective constructions for this term as it appears in claim 15.

Term	Intel's Construction	Transmeta's Construction
operating conditions internal to said computer processor [Power Management Term 5]	a plurality of types of operating conditions, excluding core utilization, that are internal to the computer processor	No construction necessary – plain and ordinary meaning If the Court decides a construction is necessary, this term means a plurality of types of operating conditions that are internal to the computer processor.

The dispute between the parties is whether the applicants disclaimed so-called "core utilization" as an internal operating condition during prosecution (Intel's proposal), or whether

any condition meets the claim language (Transmeta's proposal). Core utilization refers to how much of its processing power the computer processor is using. *See, e.g.*, Horden at 1:61-65.

During prosecution, the examiner repeatedly rejected the claims based on Horden. In response, the applicants characterized "core utilization" as an "external" consideration:

> Horden . . . actually teaches away from using operating conditions internal to a computer processor as claimed . . . . [In Horden,] the decision as to what voltage and frequency at which to operate is made by the operating system, based on considerations external to the processor. For example, the operating system identifies whether core utilization and the corresponding throughput can be handled at an idle frequency or if a higher frequency is needed . . . . In contrast, Claim 12 recites, "monitoring operating conditions internal to a computer processor" and "determining a frequency and a voltage suitable for a level of power consumption, based on said operating conditions."

JA-A at 253 ['061 FH at 7/7/03 Response, p. 20] (emphasis in original omitted). By arguing during prosecution that core utilization is not an internal operating condition, Transmeta cannot now recapture that claim scope. *See Springs Window Fashions LP v. Novo Indus., LP*, 323 F.3d 989, 995 (Fed. Cir. 2003) (prosecution argument comprises disclaimer even if unsuccessful).

3. "state" (Power Management Term 8)

Claim 30 includes a limitation regarding the "state" of the processor. The table below shows the parties' respective constructions for this term.

Term	Intel's Construction	Transmeta's Construction
a state of said computer processor [Power Management Term 8]	the activeness or idleness of the computer processor	a condition of the computer processor such as activeness or idleness

The dispute between the parties is whether this term refers to the activeness or idleness of the processor (Intel's proposal) or broadly includes any "condition" (Transmeta's proposal). Transmeta's proposal to replace the term "state" with the broader term "condition" is improper because Transmeta argued to the Patent Office that temperature is a "condition" but not a "state."

JA-A at 1032, 1036 [Reexam Response, pp. 25, 29]. Therefore, according to Transmeta, not all states are conditions. In addition, Transmeta's non-exclusive examples will not help the jury decide what is included within or excluded from the claims scope. *See* JA-C at 017 [MPEP § 2173.05(d)] ("exemplary claim language" including "for example" or "such as" may "lead to confusion over the intended scope of the claim").

4. "clock frequency generator" (Power Management Term 3)

Claims 8 and 56 require a "clock frequency generator." The table below shows the parties' respective constructions for this term.

Term	Intel's Construction	Transmeta's Construction
clock frequency generator [Power Management Term 3]	a unit that provides individual clock frequencies for each of a plurality of components including a processing unit of the processor, the system memory, and the system bus	No construction necessary – plain and ordinary meaning

The dispute between the parties is whether claims 8 and 56 are limited to the "clock frequency generator" disclosed in the specification (Intel's proposal), or also cover the prior art generator distinguished in the specification (Transmeta's position).

It is clear from the specification that the claimed generator does <u>not</u> include the prior art:

> The generator 17 . . . [produces] one or more clocks for operation of the various system memory components shown as system memory 14 in the figure, the system bus, and any other components which might utilized [sic] a separate clock.
>
> It should be specifically noted that <u>contrasted to prior art systems, the programmable frequency generator is able to provide individual frequencies selectable for each of these components</u>.

'061 at 3:20-30. It is this generator – as disclosed and distinguished from the prior art – that comprises the claimed clock frequency generator. Transmeta's position – that all clock frequency generators fall within the claims – cannot be reconciled with this clear disclaimer. *See*

SciMed, 242 F.3d at 1343 ("claims should not be read so broadly as to encompass the distinguished prior art structure").

5. "processor determining" (Power Management Terms 7 and 11)

Claims 23, 30 and 39 relate to determining a frequency and a voltage. The table below shows the parties' respective constructions for this term as it appears in claims 23 and 30.

Term	Intel's Construction	Transmeta's Construction
said computer processor determining a frequency and a voltage [Power Management Term 7]	the computer processor itself, not the operating system, determines a frequency and a voltage	No construction necessary – plain and ordinary meaning

The dispute between the parties is whether the applicants disclaimed the operating system as outside the scope of the computer processor, as Intel proposes. Transmeta contends that no construction of this term is necessary.

During prosecution, the applicants distinguished the prior art as follows:

> Horden may disclose that an operating system determines what frequency and voltage at which to operate a processor. However, this teaching fails to teach or suggest the claimed limitation of a computer processor determining the frequency and voltage, which is vastly different from an operating system performing these tasks.

JA-A at 290 ['061 FH at 2/19/04 Response, p. 15]. By unambiguously arguing that an operating system falls outside the scope of the term "computer processor," the applicants narrowed the scope of the claims. *See Std. Oil*, 774 F.2d at 452.

6. "determining . . . maximum[s and a] . . . minimum" (Power Management Term 1)

Claim 1 requires three "determining" steps as set forth in the table below.

Term	Intel's Construction	Transmeta's Construction
[1] ***determining*** *a maximum allowable power consumption level* from an operating condition of the processor, said computer processor [2] ***determining*** *a maximum frequency* which provides power not greater than *the allowable power consumption level*, said computer processor [3] **determining** a minimum voltage which allows operation at *the maximum frequency determined* [Power Management Term 1]	after determining a maximum allowable power consumption level from an operating condition of the processor, the computer processor determines, in a separate step, a maximum frequency which provides power not greater than the determined allowable power consumption level, and the computer processor determines, in another separate step, a minimum voltage which allows operation at the determined maximum frequency	based on an operating condition of the processor, the computer processor determines a maximum allowable power consumption level by determining a corresponding maximum frequency and a minimum voltage which allows operation at the maximum frequency

The dispute between the parties is whether this term includes three separate steps performed in the recited order (Intel's position), or a single omnibus step (Transmeta's position).

The sequential nature of the claim language makes clear that each step must be performed separately in the recited order. In *E-Pass,* the Federal Circuit explained that when a step of a method claim refers to "the completed results of the prior step," then the steps must be performed in the recited order. 473 F.3d at 1222; *see also Mantech Envt'l Corp. v. Hudson Envt'l Servs., Inc.*, 152 F.3d 1368, 1376 (Fed. Cir. 1998) ("the sequential nature of the claim steps is apparent from the plain meaning of the claim language").

Here, step [2] requires the computer to consider "the allowable power consumption level." Step [2] therefore must follow step [1] in which the computer determines an "allowable power consumption level." Likewise, because step [3] requires the computer to determine a

minimum voltage that allows the processor to operate at "the maximum frequency determined," step [3] must follow step [2] in which the computer determines what is the maximum frequency.

Transmeta's proposal, in contrast, blurs the separate steps into a single agglomeration. Transmeta proposes performing step [1] "by" performing a combination of steps [2] and [3]. The Court should reject Transmeta's improper attempt to excise step [1] from the claim and combine steps [2] and [3], because "claims are interpreted with an eye toward giving effect to all terms in the claim." *Bicon*, 441 F.3d at 950.

7. "determining a frequency and a voltage" (Power Management Terms 2 and 6)

Claims 1, 15, 23, 30, and 39 require the computer to determine a frequency and a voltage. The table below shows the parties' constructions for this term.

Term	Intel's Construction	Transmeta's Construction
determining a frequency and a voltage [Power Management Term 6]	determine a frequency and a voltage based at least on analyzing commands to be executed by the processor	No construction necessary – plain and ordinary meaning

The dispute between the parties is whether the commands to be executed by the processor must be analyzed as part of the frequency/voltage determination, as Intel proposes. Transmeta contends that no construction of this term is necessary.

The "Summary of the Invention" section explains that the "present invention is realized by . . . changing the operating characteristics of the central processor to a level commensurate with the operations being conducted." '061 at 2:16-24; *see also id.* at Abstract. The invention, therefore, is directed to increasing the frequency/voltage when doing so is necessary to perform the required operations and decreasing the frequency/voltage when a higher frequency/voltage is not necessary to perform the required operations.

Because the claims require "determining" the appropriate voltage and frequency for the processor, an analysis of some kind must be performed. Although the patent discloses multiple

methods for analyzing when to decrease the voltage and frequency, the only method disclosed for analyzing when to increase the voltage and frequency is by analyzing the commands to be executed by the processor. *Id.* at 5:47-50; 7:40-45. Therefore, in order for the processor to determine whether to increase or decrease the voltage and frequency, it must first analyze the commands to be executed. *See On Demand*, 442 F.3d at 1340 ("the claims cannot be of broader scope than the invention that is set forth in the specification").

8. "means for detecting . . . and causing" (Power Management Term 4)

The parties agree that this term in claim 8 is subject to §112, ¶ 6 and that the claimed function is "detecting the values indicative of operating conditions . . . and causing the power supply and clock frequency generator to furnish an output clock frequency and voltage level . . . and to generate concurrently frequencies . . . ." The parties also agree that the corresponding structure includes control software. The dispute is whether the corresponding structure also includes "a set of registers in the processor – such as the clock divider register 22," as Transmeta proposes. Transmeta's attempt to include this structure is improper.

First, there is no link between the registers described in the patent and the recited function. *See Default Proof* 412 F.3d at 1297-98. Only the control software is linked to the recited function. *See* '061 at 5:15-20; 2:64-3:12. Second, the registers described in the patent (including the clock divider register 22) do not "actually perform the recited function" but instead, at most, "merely enable the pertinent structure to operate as intended." *Asyst Techs., Inc. v. Empak, Inc.*, 268 F.3d 1364, 1371 (Fed. Cir. 2001); '061 at 4:29-32 ("In order to allow the master control unit 18 to accomplish these operations, the processing unit 16 includes a number of registers which are utilized by the control software and the hardware."). As explained in *Asyst*, "An electrical outlet enables a toaster to work, but the outlet is not for that reason considered part of the toaster." 268 F.3d at 1371.

II. THE INTEL PATENTS-IN-SUIT

As with the Transmeta Patents-in-Suit, Transmeta's proposed constructions do not properly account for the true inventions, as described in the patent specifications. Here, however, rather than ignoring disclaimers or descriptions in the specifications of what the applicants considered to be their inventions (as Transmeta has done for its own asserted patents), Transmeta's proposed constructions for the Intel Patents-in-Suit consistently, and improperly, limit the claims to examples described in the specifications. The law is clear that it is appropriate to rely on the manner in which the patentee uses a term in the specification to determine whether it was intended that the claims and the description in the specification are intended to be coextensive. *Phillips*, 415 F.3d at 1323. It is also appropriate to limit the claims to an embodiment described in the specification where there is nothing in the context of the patent to indicate that the patentee contemplated any alternative. *Id.* But it is improper to limit the scope of claims to a example in the specification of how to practice the invention, as Transmeta seeks to do with respect to the Intel patents. *Id.*

A. THE INTEL POWER MANAGEMENT PATENT

1. Introduction To The Intel Power Management Patent

The '375 patent – cited prior art to Transmeta's '061 Power Management patent discussed above – is directed to reducing power consumption in a processor or other electronic device by dynamically scaling voltage and frequency (*i.e.*, changing both voltage and frequency to manage power consumption). The '375 invention improves energy conservation by changing voltage and frequency – based on conditions such as idle time and temperature – while the processor is performing its regular operations. '375 at 1:55-2:4; 2:20-27; 2:34-42.

2. The "supplies to" and "provides to" terms (Intel Power Management Terms 2 and 4)

All of the asserted claims of the '375 patent recite that "a clock signal having a scalable frequency" be "supplie[d]" to the processor or other electronic device and/or that "a power supply signal having a scalable voltage" be "provide[d]" to processor or other electronic device. The table below shows the parties' respective constructions for these terms.

Term	Intel's Construction	Transmeta's Construction
"supplies a clock signal having a scalable frequency to" [Intel Power Management Term 2]	supplies a clock signal having a frequency that can be increased or decreased as necessary to	supplies to, from an external source
"provides a power supply signal having a scalable voltage to" [Intel Power Management Term 4]	provides [a power supply signal having a voltage that can be increased or decreased as necessary] to	provides to, from an external source

The parties agree that it is "a clock signal having a frequency that can be increased or decreased as necessary" that is "supplied" to the device, and that it is "a power supply signal having a voltage that can be increased or decreased as necessary" that is "provided" to the device. The dispute between the parties is whether these signals may be supplied and provided by the device itself or an external source (Intel's position), or must be supplied and provided by an external source (Transmeta's position).

Neither the claim language nor the intrinsic record require that these signals be supplied and provided by a source external to the claimed device. Transmeta's proposed construction should be rejected as improperly limiting the claims to an "example of how to practice the invention" shown in Figure 3. *See Phillips*, 415 F.3d at 1323; *see also Gart v. Logitech, Inc.*, 254 F.3d 1334, 1342 (Fed. Cir. 2001) ("drawings are not meant to represent 'the' invention or to

limit the scope of coverage defined by the words used in the claims themselves"). Nothing in the specification or prosecution history limits the claimed invention as Transmeta proposes.

3. "electronic device" (Intel Power Management Term 1)

Claims 1, 30, and 33 are directed to an "electronic device." The table below shows the parties' respective constructions for this term.

Term	Intel's Construction	Transmeta's Construction
"electronic device" [Intel Power Management Term 1]	No construction necessary	a self-contained electronic component

The dispute between the parties is whether the patent limits the claimed device to a "self-contained" device, as Transmeta proposes. Intel's position is that no construction is necessary.

The patent explains that an example of an "electronic device" is a computer processor. *Id.* at 4:5-6 ("an electronic device (*e.g.*, a microprocessor)"). There is no support in the claim language or the intrinsic record for Transmeta's proposed limitation. Moreover, it is unclear what Transmeta's proposed construction even means, and will leave the jury guessing, as to whether, for example, a computer plugged into a wall socket – which is a disclosed embodiment (*id.* at 6:27-9) – would satisfy this limitation. *See Chimie*, 402 F.3d at 1377 ("As we have frequently stated, a construction that would not read on the preferred embodiment would rarely, if ever be correct") (quotations omitted).

4. "event" (Intel Power Management Term 6)

Claims 1 and 16 recite that the signals used to initiate the voltage and frequency scaling are generated in response to an "event." The table below shows the parties' respective constructions for "event."

Term	Intel's Construction	Transmeta's Construction
"event" [Intel Power Management Term 6]	circumstance warranting a change in power consumption	a temperature related occurrence

The dispute between the parties is whether any circumstance warranting a change in power consumption may trigger voltage and frequency scaling (Intel's position) or only a "temperature related occurrence" can do so (Transmeta's position).

Neither the claim language nor the intrinsic record limits the claimed invention to voltage and frequency scaling only to address a "temperature related occurrence," as Transmeta proposes. To the contrary, even the abstract expressly states that the invention dynamically scales voltage and frequency in response to conditions other than temperature, including the "idle" time of the electronic device. *See, e.g.*, '375 at Abstract.

Similarly, the "Summary of the Invention" section states that the claimed invention dynamically scales voltage and frequency in response to "at least one of two conditions," and it identifies as exemplary events both a "temperature related occurrence (*i.e.*, when the temperature exceeds a "thermal band") and a non-temperature related event (*i.e.*, "where the electronic device is detected to be idle for a selected percentage of power-on time"). '375 at 2:36-42; *see also* 4:19-25 (conditions triggering voltage and frequency scaling may include when "the temperature of the CPU 110 exceeds a thermal band or the CPU 110 is experiencing excessive idle time"). Additional events triggering voltage and frequency scaling are also disclosed. *See, e.g., id.* at 5:46-50 ("the event that the HCD storage element is set frequently"); *id.* at 6:45-8 ("If the hardware product . . . is in De-turbo mode"); *id.* at 6:60-4 ("when DLNT mode is enabled"); *id.* at 6:35-9 ("if the electronic device is receiving power from the battery power supply"). None of these events is a "temperature related occurrence."

5. "thermal band" (Intel Power Management Term 8)

Dependent claims 3, 4, 18, 19, and 31 relate to dynamically scaling voltage and frequency when the temperature exceeds a "thermal band." The table below shows the parties' respective constructions for "thermal band."

Term	Intel's Construction	Transmeta's Construction
"thermal band" [Intel Power Management Term 8]	acceptable operating temperature range defined by upper (maximum) and lower (minimum) temperature limits	operating temperature range defined by upper (maximum) and lower (minimum) temperature limits

The parties agree that the "thermal band" is a temperature range but dispute whether it defines an "acceptable" temperature range (Intel's position) or any temperature range within which the device remains operational (Transmeta's position).

The specification describes two types of thermal bands: an "absolute" band requiring "immediate device shutdown if exceeded" and an acceptable operational band beyond which "throttling" (*i.e.*, decreasing voltage and frequency to reduce temperature) occurs. '375 at 4:25-30. Because the claims at issue are directed to changing the voltage and frequency in response to exceeding the thermal band, the "thermal band" at issue in these claims is the acceptable operational band. The "absolute" band cannot be the maximum temperature of the "thermal band," as Transmeta proposes, because the claimed temperature-induced voltage and frequency scaling does not begin until after the device temperature "exceeds" the thermal band, and exceeding the "absolute" band requires "immediate device shut-off." *Id.* Transmeta's attempt to read the invention out of the claim language is nonsensical and should be rejected.

B. THE INTEL ADDRESS TRANSLATION PATENTS

1. Introduction To The Intel Address Translation Patents

The '554 and '605 patents – prior art to the Belgard patents discussed above (and cited to the Patent Office during the prosecution of one of them) – are directed to managing different sized memories and pages within memories. '554 at 3:30-48. Because the number of locations in the physical memory is limited, performance may also be limited. *Id.* at 1:16-26. The claimed invention improves performance and flexibility by allowing the computer to select different sized physical addresses and different sized "pages." *Id.* at 3:30-47.

2. "linear address" (Intel Address Translation Term 1)

All of the asserted claims are directed to translating "linear addresses" into "physical addresses." The table below shows the parties' respective constructions for "linear address."

Term	Intel's Construction	Transmeta's Construction
"linear address" [Intel Address Translation Term 1]	a logical address having a fixed size and that translates into an actual physical address	an address identifying a location in a continuous unsegmented address space, which is translated from a virtual address, and which is translated into a physical address

The dispute between the parties is whether a "linear address" must (1) be translated from a "virtual address," and (2) identify a location in an "unsegmented address space," as Transmeta proposes. Intel's position is that there is no support in the claim language or the intrinsic record for either of these proposed limitations, both of which require the exemplary "segmentation" process described in the specification, which translates "virtual addresses" into "linear addresses." It is clear from the patents that the segmentation process is exemplary only:

> In the preferred embodiment, the translator translates a linear address supplied by a microprocessor in a computer that uses segmentation methods to obtain the linear address by translating a virtual address. However, it should be apparent to one skilled in the art that, in other computer architectures, the translator of the present invention could translate a computer supplied address obtained by another method.

'554 Patent at 3:52-59. Transmeta's construction should be rejected as improperly limiting the claims to an "example of how to practice the invention." *See Phillips*, 415 F.3d at 1323.

3. "physical address" (Intel Address Translation Term 4)

As explained above, all of the asserted claims are directed to translating "linear addresses" into "physical addresses." The table below shows the parties' respective constructions for "physical address."

Term	Intel's Construction	Transmeta's Construction
"physical address" [Intel Address Translation Term 4]	address that is sufficient to unambiguously specify the location of a desired unit of data equal in size to the smallest storage location addressable by the processor, typically one byte	a location in the computer's physical, i.e., real memory

The dispute between the parties here is the same dispute between the parties with respect to the Belgard patents discussed above: whether a physical address unambiguously specifies the location of the smallest unit of data (Intel's position) or any location in the computer's physical memory (Transmeta's position). As explained above, a "physical address" must be sufficient to unambiguously specify the location of the smallest unit of data stored therein because the physical address must precisely address every byte of data. *See, e.g.*, '554 at 7:52-56.

4. The "paging" terms

The asserted claims of the '554 patent and claim 1 of the '605 patent are directed to performing "paging." The table below shows the parties' respective constructions for "paging" and "page frame size." "Page frame" involves the same dispute as "page frame size."

Term	Intel's Construction	Transmeta's Construction
"paging" [Intel Address Translation Term 5]	using fixed sized blocks of memory to translate logical addresses into physical addresses	using blocks of memory of predetermined size to translate logical addresses into physical addresses
"page frame size" [Intel Address Translation Term 7]	fixed size of the memory unit used in "paging"	the size of a contiguous aligned block of physical memory

(a) "paging" (Intel Address Translation Term 5)

The parties agree that "paging" refers to using blocks of memory to translate logical addresses into physical addresses, but disagree as to whether the blocks are of a fixed-size (Intel's position) or a predetermined size (Transmeta's position). Only Intel's construction is supported by the specification, which defines "pages" as "fixed sized blocks of memory," ('554

at 2:4-6), and does not limit pages to a "predetermined size," as Transmeta proposes. Rather, the claimed invention includes using "multiple page sizes." *Id.* at 4:57-9; claim 19.

(b) "page frame" and "page frame size" (Intel Address Translation Terms 6 and 7)

The dispute between the parties is whether "page frame" and "page frame size" are limited to a "contiguous aligned" block of memory, as Transmeta proposes. Intel's position is that there is no support in the claim language or the intrinsic record for this limitation, which Transmeta improperly seeks to import from the cited prior art. Because the specification clearly defines the "page frame" as the unit of physical memory resulting from paging (*id.* at 2:5-9), it is improper to rely on the prior art to construe this term. *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1584 (Fed. Cir. 1996) ("reliance on [prior art] is unnecessary, and indeed improper, when the disputed terms can be understood from a careful reading of the public record").

5. "control unit" and "paging unit" (Intel Address Translation Terms 2 and 3)

All of the asserted claims of the '554 patent and claim 1 of the '605 patent recite a "control unit" storing an "indication" and a "paging unit" that uses the indication to perform paging. The table below shows the parties' respective constructions for these terms.

Term	Intel's Construction	Transmeta's Construction
"control unit" [Intel Address Translation Term 2]	No construction necessary	control circuitry including registers within the microprocessor
"paging unit" [Intel Address Translation Term 3]	a unit to perform "paging"	circuitry within the microprocessor used in paging

The dispute between the parties is whether these units are limited to hardware circuitry, as Transmeta proposes. Intel's position is that there is no support in the claim language or the intrinsic record to limit the term "unit" in this way. In fact, the specification discloses using

"conventional logic," which was well-known in the art to refer to both software and hardware implementations and combinations thereof. '554 at 6:57-61; 11:43-54.

C. Intel Multimedia Processing Patents

1. Introduction To The Intel Multimedia Processing Patents

The Intel multimedia processing patents are directed generally to performing computer operations on multiple data elements in parallel. The problem addressed by these patents is that multimedia computer applications such as image processing and video compression "often manipulate large amounts of data where only a few bits are important," and therefore "waste" the processing and storage capacity of the processor. *See*, *e.g.*, '101 at 1:38-49. To address this problem, each of the multimedia processing patents is directed to a new computer instruction which increases efficiency by performing the same operation on parallel data elements.

2. "Packed Data" (Intel Multimedia Term 1)

All of the asserted claims of the '101, '275, '634, and '529 patents are directed to performing operations on "packed data." The table below shows the parties' respective constructions for "packed data."

Term	Intel's Construction	Transmeta's Construction
"packed data" [Intel Multimedia Term 1]	unit of data that consists of a plurality of data elements of the same size	unit of data that is fully populated with a plurality of data elements of the same size

The parties agree that "packed data" refers to a unit including a plurality of data elements of the same size, but disagree as to whether the data elements must "fully populate" the unit as Transmeta proposes. Intel's position is that there is no support in the claim language or the intrinsic record for this limitation. In fact, Transmeta's proposal is inconsistent with an embodiment described in the specification in which a 64-bit packed data is stored in a register capable of storing 80-bit values. *See* '634 at 9:11-18 ("all eighty bits are used for storing floating

point data, while only sixty-four are used for packed data"); *see also* '101 at 4:45-52. In this embodiment, the packed data clearly does not fully populate the unit as Transmeta proposes.

3. "copying" and "decoding" (Intel Multimedia Terms 2 and 4)

All of the asserted claims of the '101 and '275 patents include the step of "copying" data elements, and all of the asserted claims of the '101 and '275 patents, and claims 30 and 36 of the '529 patent, include the step of "decoding" the claimed instructions. The table below shows the parties' respective constructions for "copying" and "decoding."

Term	Intel's Construction	Transmeta's Construction
"copying" [Intel Multimedia Term 4]	No construction necessary	bitwise replication independent of the value of the data
"decoding" [Intel Multimedia Term 2]	No construction necessary	transforming an external representation of an instruction into internal operations or commands

The dispute between the parties is whether these term require construction. Intel's position is that these are simple terms unlikely to confuse the jury. Transmeta's proposed constructions, on the other hand, are confusing and find no support in the claim language.

4. " . . . two, four, or eight data elements" (Intel Multimedia Term 5)

Dependent claims 4 and 12 of the '101 patent and dependent claims 3 and 8 of '275 patent recite that the claimed packed data each include "either two, four, or eight data elements." The table below shows the parties' respective constructions for this term.

Term	Intel's Construction	Transmeta's Construction
"each includes either two, four, or eight data elements" [Intel Multimedia Term 5]	No construction necessary	the computer system has the capability of manipulating each of the specified alternatives

The dispute between the parties is whether the meaning of this term is apparent on its face, *i.e.*, the packed data <u>each</u> include either two, four, or eight data elements (Intel's position)

or this term should be construed to require the computer to be capable of manipulating all three of these alternatives (Transmeta's proposal). It is clear from the claim language itself, as well as the specification, that the computer need only be capable of manipulating any one of these alternative numbers of data elements. *See, e.g.*, '101 at 7:13-31.

5. The "intermediate" terms (Intel Multimedia Terms 6, 7, and 8)

The asserted claims of the '634 patent refer to "intermediate data elements" and "intermediate results" (also referred to as "intermediate result data elements"). The table below shows the parties' respective constructions for "intermediate result." The other "intermediate" terms involve the same dispute.

Term	Intel's Construction	Transmeta's Construction
"intermediate result" [Intel Multimedia Term 8]	No construction necessary	a result that is a fully calculated product

The dispute between the parties is whether the meaning of this term is apparent on its face, *i.e.*, the intermediate stage of the instruction (Intel's position) or this term should be construed to require a "fully calculated product" (Transmeta's proposal). Neither the claim language nor the intrinsic record requires that the intermediate stage of the instruction be a "fully calculated product," as Transmeta proposes. Moreover, Transmeta's proposed construction is inconsistent with the specification, which describes intermediate results as "temporary" and distinguishes intermediate results from the final result. *See, e.g.*, '634 at 6:60-62.

CONCLUSION

For the foregoing reasons, Intel respectfully requests that the Court adopt its proposed constructions for the disputed terms of the Transmeta Patents-in-Suit, and its proposed constructions for the disputed terms of the Intel Patents-in-Suit.

Respectfully submitted,

/s/ *Karen L. Pascale*

John W. Shaw (No. 3362)
Karen L. Pascale (No. 2903)
YOUNG CONAWAY STARGATT & TAYLOR, LLP
The Brandywine Building
1000 West Street, 17th Floor
Wilmington, DE 19801
(302) 571-6600
kpascale@ycst.com

Attorneys for Defendant Intel Corporation

OF COUNSEL:

Matthew D. Powers
Jared Bobrow
Steven S. Cherensky
WEIL, GOTSHAL & MANGES LLP
201 Redwood Shores Parkway
Redwood Shores, CA 94065
(650) 802-3000

Kevin Kudlac
WEIL, GOTSHAL & MANGES LLP
8911 Capital of Texas Highway, Suite 1350
Austin, TX 78759
(512) 349-1930

Dated: October 19, 2007

CERTIFICATE OF SERVICE

I, Karen L. Pascale, Esquire, hereby certify that on October 19, 2007, I caused to be electronically filed a true and correct copy of the foregoing document with the Clerk of the Court using CM/ECF, which will send notification that such filing is available for viewing and downloading to the following counsel of record:

Jack B. Blumenfeld [jbbefiling@mnat.com]
Karen Jacobs Louden [kjlefiling@mnat.com]
MORRIS, NICHOLS, ARSHT & TUNNELL LLP
1201 North Market Street
P.O. Box 1347
Wilmington, DE 19899
(302) 658-9200

I further certify that on October 19, 2007, I caused a copy of the foregoing document to be served on the above-listed counsel and on the following non-registered participants in the manner indicated:

By E-Mail and Hand Delivery

Jack B. Blumenfeld [jblumenfeld@mnat.com]
Karen Jacobs Louden [klouden@mnat.com]
MORRIS, NICHOLS, ARSHT & TUNNELL LLP
1201 North Market Street
P.O. Box 1347
Wilmington, DE 19899
(302) 658-9200

By E-Mail

Robert C. Morgan [robert.morgan@ropesgray.com]
Laurence S. Rogers [laurence.rogers@ropesgray.com]
Steven Pepe [steven.pepe@ropesgray.com]
ROPES & GRAY LLP
1211 Avenue of the Americas
New York, NY 10036

Norman H. Beamer [norman.beamer@ropesgray.com]
ROPES & GRAY LLP
525 University Avenue
Palo Alto, CA 94301

YOUNG CONAWAY STARGATT & TAYLOR LLP

/s/ Karen L. Pascale

John W. Shaw (No. 3362) [jshaw@ycst.com]
Karen L. Pascale (No. 2903) [kpascale@ycst.com]
The Brandywine Building
1000 West Street, 17th Floor
P.O. Box 391
Wilmington, Delaware 19899-0391
Attorneys for Intel Corporation